

SANDIA REPORT

SAND93-1641 • UC-706

Unlimited Release

Printed May 1994

MICROFICHE

A Three-Dimensional Fast Solver for Arbitrary Vorton Distributions

James H. Strickland, Roy S. Baty

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-84AL85000

Approved for public release; distribution is unlimited.



8635035

**SANDIA NATIONAL
LABORATORIES
TECHNICAL LIBRARY**

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A04
Microfiche copy: A01

A Three-Dimensional Fast Solver for Arbitrary Vorton Distributions

James H. Strickland
Roy S. Baty
*Parachute Technology & Unsteady Aerodynamics Dept.
Organization 1552
Sandia National Laboratories
Albuquerque, New Mexico 87185*

Abstract

A method which is capable of an efficient calculation of the three-dimensional flow field produced by a large system of vortons (discretized regions of vorticity) is presented in this report. The system of vortons can, in turn, be used to model body surfaces, container boundaries, free-surfaces, plumes, jets, and wakes in unsteady three-dimensional flow fields. This method takes advantage of multipole and local series expansions which enables one to make calculations for interactions between groups of vortons which are in well-separated spatial domains rather than having to consider interactions between every pair of vortons. In this work, series expansions for the vector potential of the vorton system are obtained. From such expansions, the three components of velocity can be obtained explicitly. A Fortran computer code FAST3D has been written to calculate the vector potential and the velocity components at selected points in the flow field. In this code, the evaluation points do not have to coincide with the location of the vortons themselves. Test cases have been run to benchmark the truncation errors and CPU time savings associated with the method. Non-dimensional truncation errors for the magnitudes of the vector potential and velocity fields are on the order of 10^{-4} and 10^{-3} respectively. Single precision accuracy produces errors in these quantities of up to about 10^{-5} . For less than 1,000 to 2,000 vortons in the field, there is virtually no CPU time savings with the fast solver. For 100,000 vortons in the flow, the fast solver obtains solutions in 1% to 10% of the time required for the direct solution technique depending upon the configuration.

(this page intentionally left blank)

Nomenclature

A_n^m	=	coefficient in multipole expansion
\tilde{A}_j^k	=	coefficient in shifted multipole expansion
B_j^k	=	coefficient in local expansion
\tilde{B}_j^k	=	coefficient in shifted local expansion
D_1	=	domain 1
D_2	=	domain 2
D_3	=	domain 3
D_4	=	domain 4
D_F	=	field domain
D_S	=	source domain
D_n^m	=	coefficient defined by Equation 11
\vec{e}_s	=	unit vector in the direction of Δs
i	=	index or $\sqrt{-1}$
Ja_n^m	=	coefficient defined by Equation 10
$Jab_m^{n,k}$	=	coefficient defined by Equation 15
$Jb_{n,m}^k$	=	coefficient defined by Equation 18
j	=	index
k	=	index
l	=	index or box level
m	=	index
mh	=	limits defined by Equation 12
ml	=	limits defined by Equation 12
N	=	order or precision of multipole and local expansions
N_E	=	total number of evaluation points in the field
N_F	=	maximum number of field evaluation points in a box
N_S	=	maximum number of vortons in a box
N_V	=	total number of vortons in the field
N_{Vc}	=	$N_F = N_S = N_{Vc}$ for all benchmark tests

Nomenclature (continued)

n	=	index
P	=	field point
P_n^m	=	associated Legendre function
q	=	variable defined in Equation 46
R	=	ring radius
R_F	=	radius of the field domain
R_S	=	radius of the source domain
r	=	radius in spherical coordinates
S_l	=	box dimension at level l
t_{cpu}	=	CPU time
x	=	cartesian coordinate, argument of associated Legendre function
x_j	=	cartesian coordinate in the j^{th} direction
y	=	cartesian coordinate
z	=	cartesian coordinate
U_s	=	self induced velocity
\vec{U}	=	velocity vector
\vec{U}_1	=	velocity vector from list 1 boxes
\vec{U}_3	=	velocity vector from list 3 boxes
\vec{U}_{24}	=	velocity vector from list 2 and 4 boxes
U^+	=	non-dimensional velocity
$ U _{ei}$	=	magnitude of velocity vector at point i using direct method
$ U _i$	=	magnitude of velocity vector at point i using fast method
Y_n^m	=	spherical harmonic defined by Equation 4
α	=	angle in spherical coordinates or constant in Equation 47
α_i	=	angle in spherical coordinates associated with i^{th} vorton
β	=	angle in spherical coordinates or function defined by Equation 50
β_i	=	angle in spherical coordinates associated with i^{th} vorton
ε	=	constant in Equation 39
ε_U	=	truncation error in the velocity vector defined by Equation 54

Nomenclature (continued)

ε_Φ	=	truncation error in the vector defined by Equation 54
Γ	=	circulation strength
$(\Gamma\Delta s)_{l,i}$	=	magnitude of $\omega\Delta v$ in the l^{th} direction of the i^{th} vorton
$(\Gamma\Delta s)_j$	=	magnitude of $\omega\Delta v$ in the j^{th} direction
Δs	=	vorton length, arc length
Δv	=	incremental volume
ξ_U	=	core function for the velocity vector
ξ_Φ	=	core function for the vector potential
θ	=	angle in spherical coordinates (see Figure 3)
ρ	=	radius in spherical coordinates or ratio of radius r to core radius σ
ρ_i	=	radius to i^{th} vorton
σ	=	core radius of vorton
Φ_l	=	l^{th} component of $\vec{\Phi}$
$\vec{\Phi}$	=	vector potential
$\vec{\Phi}_1$	=	vector potential from list 1 boxes
$\vec{\Phi}_3$	=	vector potential from list 3 boxes
$\vec{\Phi}_{24}$	=	vector potential from list 2 and 4 boxes
Φ^+	=	non-dimensional potential
Φ_s^+	=	self induced value of Φ^+
$ \Phi _{ei}$	=	magnitude of the vector potential at point i using direct method
$ \Phi _i$	=	magnitude of the vector potential at point i using fast method
ϕ	=	angle in spherical coordinates (see Figure 3)
$\vec{\omega}$	=	vorticity vector
ω	=	magnitude of $\vec{\omega}$

(this page intentionally left blank)

Table of Contents

1	INTRODUCTION	11
1.1	Motivation for Work	11
1.2	General Methodology	11
2	SOURCE AND FIELD-POINT EXPANSIONS	15
2.1	Multipole Expansion About the Source Domain Center	15
2.2	Translation of the Source Domain Center	17
2.3	Local Expansion About the Field Domain Center	18
2.4	Translation of the Field Domain Center	18
2.5	Local Series Expansion Due to Single Vortons	20
3	METHOD OF SOLUTION	21
3.1	Adaptive Mesh	21
3.1.1	Mesh Generation	21
3.1.2	Box Lists	22
3.2	Generation of Series Coefficients	26
3.2.1	Multipole Coefficients	26
3.2.2	Local Coefficients	26
3.3	Calculation of Field Variables	26
3.3.1	Vector Potential	27
3.3.2	Velocity	27
4	CORE FUNCTION AND SELF INDUCTION	32
4.1	Core Functions	32
4.2	Self Induction	34
4.2.1	Velocity	34
4.2.2	Potential	35
5	BENCHMARK TESTS	37
5.1	General Description of Benchmark Tests	37
5.2	Results From Cubic Configuration	38
5.3	Results From 10:1:1 Parallelepiped Configuration	42
5.4	Results From Separated Cube Domain Configuration	43
5.5	Lessons Learned form Benchmark Test Results	45
6	SUMMARY	47
7	REFERENCES	48
	APPENDIX	49
A.1	Data for Cubic Configuration	49
A.2	Data for 10:1:1 Parallelepiped Configuration	52
A.3	Data for Separated Cube Domain Configuration	56

(this page intentionally left blank)

1 INTRODUCTION

1.1 Motivation for Work

Motivation for the present work results from a need to be able to perform fast yet accurate calculations of the three-dimensional flow field produced by large numbers of vortons. A vorton is simply a mathematical representation of a discretized region of vorticity. Vortons are often referred to in the literature as “vortex blobs.” Such systems of vortons can be used, in turn, to model body surfaces, container boundaries, free-surfaces, plumes, jets, and wakes in unsteady flow fields. Simulations for most of these flows require that the flow structure evolve as a function of time, even for “steady flow” calculations. Large numbers N_V of vortons (1,000 to 100,000) are required for such simulations. Since the flow must be convected as part of the evolution process, the velocity perturbations from all vortons on all other vortons must be calculated at each time step. This requires an amount of work which is proportional to N_V^2 for each time step using direct calculation methods. The goal of this project was to reduce the amount of work to N_V or even $N_V \ln N_V$ in order to save significant CPU time in large simulations.

1.2 General Methodology

The general method comes from work done by Greengard [1] in which an algorithm to calculate potential fields for a large system of particles was developed. In that work, it was shown that the asymptotic CPU time was of order N_V for two-dimensional problems. A break-even point on the order of 100 particles was achieved for single precision accuracy for a uniform field in a square box. In an axisymmetric fast multipole method developed by Strickland and Amos [2] a break-even point on the order of 100 particles was achieved for slightly less accuracy. This was contrasted to CPU times of order N_V^2 when traditional methods were used. On the other hand, Greengard found that for particles placed uniformly in a cube, the break-even point did not occur until one had about 16,000 particles. A subsequent improvement to the algorithm by Greengard and Rokhlin [3] reduced the break even point to about 1,500 particles. These methods take advantage of multipole and local series expansions which enables one to make calculations for interactions between groups of particles which are in well-separated spatial domains rather than having to consider interactions between every pair of particles. In the present work, the spatial domains in which vortons are located are obtained in an adaptive manner which is a three-dimensional extension to the two-dimensional scheme used by Carrier, Greengard, and Rokhlin [4]. Also in the present work, the locations of the source particles (vortons) need not be the same as the locations of the field evaluation points. The present work is also complicated by the fact that the potential associated with a vorton field is a vector, not a scalar. In addition, one needs to obtain the velocity vector at a point in the flow field. This is done by taking the curl of the vector potential which requires differentiation of the series expansions for the vector potential.

As indicated, the present method takes advantage of source point (multipole) and field evaluation point (local) series expansions which enables one to make calculations for interactions between groups of vortons and field evaluation points which are in well-separated spatial domains rather than having to consider the interaction of every vorton on every evaluation point. For instance, consider the two domains D_S and D_F which are depicted in Figure 1.

Domain D_S contains a set of vortons which are acting upon (producing perturbation velocities at) a set of evaluation points contained in domain D_F . In some cases, the set of evaluation points in domain D_F might also represent the centers of a second set of vortons. In other cases they might simply represent an arbitrary set of evaluation points.

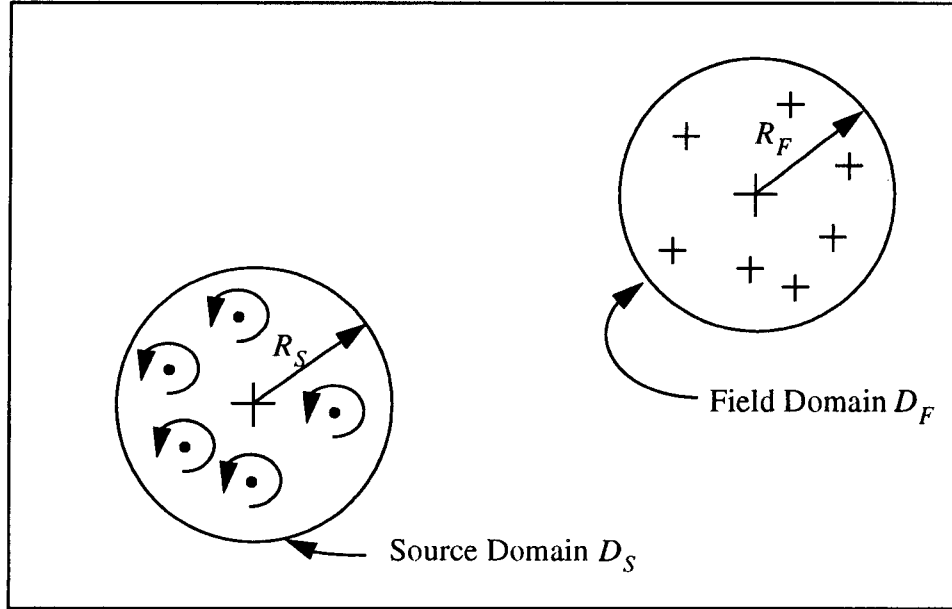


Figure 1. Source and Field Point Domains

In order to implement this method, a multipole series expansion for the vector potential about the center of the source domain D_S is first obtained. This expansion includes all of the vortons which are inside D_S . The series converges for field points which are outside of D_S . In order to be somewhat conservative, the expansion about the center of the source domain is used only for field points which are separated by at least R_S from the boundary of the source domain. Thus, for field points which are within a sphere of radius $2R_S$ with respect to the source domain center, the influence of vortons in domain D_S on the field point is computed directly. The center of the source domain may be shifted without actual evaluation of the multipole series in order to allow one to form expansions efficiently for larger source domains which contain D_S .

A local or Taylor series expansion for the vector potential can also be obtained about the center of the field domain D_F . The series converges for field points which are inside D_F . The expansion about the center of the field domain is used only for source points which are separated by at least R_F from the boundary of the field domain. Thus, for source points that are within a sphere of radius $2R_F$ with respect to the field domain center, the influence of any vorton in domain D_S on a field point is computed directly. The center of the field point domain may also be shifted without actual evaluation of the local series which allows one to form expansions for smaller field point domains contained in D_F .

In summary, the method allows one to calculate efficiently the influence of all of the vortons in D_S at all of the points in D_F when the two domains are “well separated” as defined above. There is additional efficiency inherent in the method if one exploits the possibility of moving the centers of expansion without actually having to evaluate the multipole or local series.

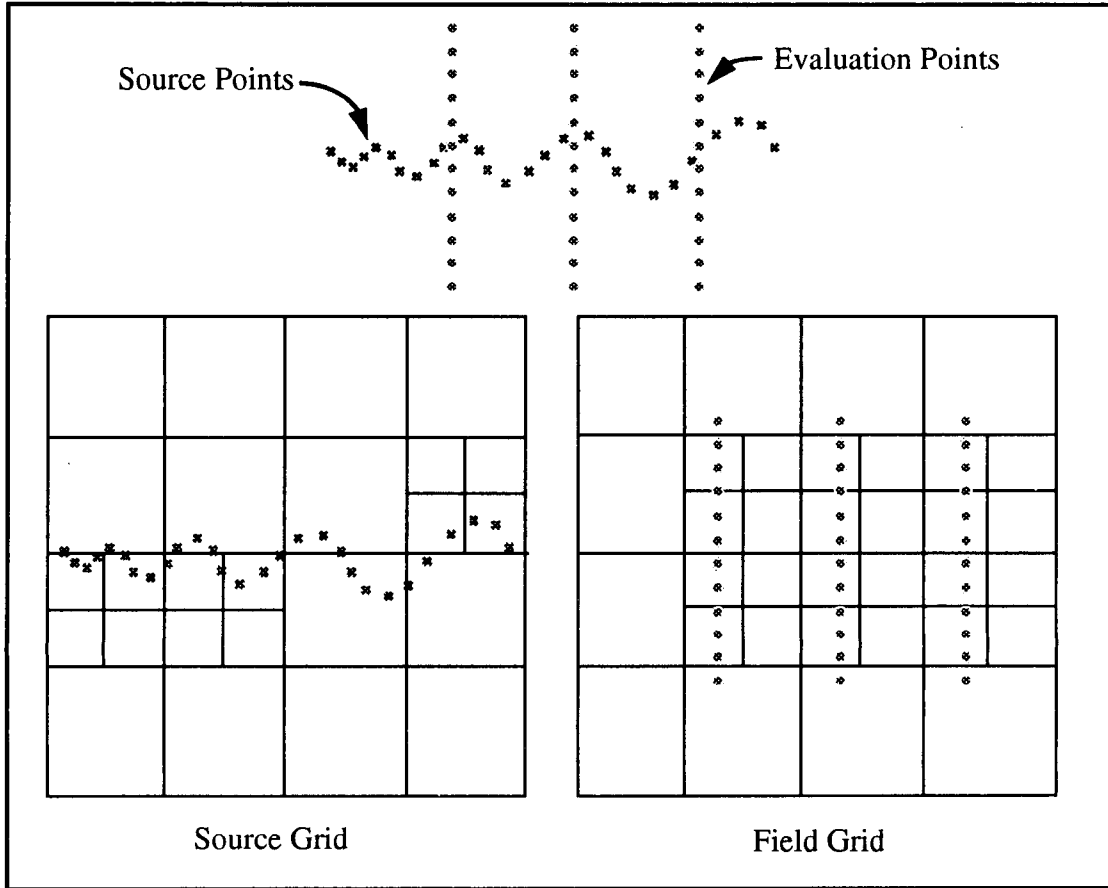


Figure 2. Adaptive Domain Mesh Scheme

The method used to divide the flow field into domains is an extension to that used by Carrier, Greengard, and Rokhlin [4] for two-dimensional fields. The method requires that an adaptive domain mesh be generated that encloses all the vorticity and field evaluation points in the flow. A cubical box is first constructed which encloses all of the vortons in the flow as well as all of the field evaluation points. This box is then subdivided into eight equal boxes. Unlike any previous scheme, the source (vorton) and field (evaluation) points are considered separately in subsequent steps. If any of the eight boxes contain more than a specified number of vortons, that particular box is subdivided into eight more boxes. This process is continued until all of the boxes contain less than the specified number of vortons. The grid for the field evaluation points is generated in a similar manner. Figure 2 illustrates generation of the source and field grid. Each box size is associated with a box level with the original box level being equal to 0. At each level, empty boxes are ignored. The ability to ignore regions of the flow-field in which vortons or evaluation points are not present is a feature of the method

which provides significant time savings. Boxes which can be subdivided are parent boxes and those which cannot are childless boxes.

A somewhat involved procedure is used to define the separation condition between a particular field box and each of the source boxes in the flow. This procedure determines the way in which vortons in a particular source box influence the vector potential at evaluation points in a particular field box. In general, each field box at each level has five possible relationships with each source box in the mesh. A formal description of these “box lists” will be given in a subsequent section. In general, the five lists produce the following types of restrictions on the use of the series expansions:

- Direct calculations must be made. Multipole and local series expansions cannot be used.
- Both multipole and local series expansions can be used.
- Multipole series expansions can be used, local series expansions cannot.
- Local series expansions can be used, multipole expansions cannot.
- Contributions from distant source boxes reside in the parent of the field box.

It should be pointed out, that the adaptive domain meshes which are generated here are not to be confused with the adaptive meshes which are generated for conventional CFD formulations. In the present case, the meshes represent a domain decomposition which is very simple when compared to meshes generated for conventional CFD formulations. Mesh generation for conventional CFD formulations is for the purpose of writing difference or element equations which satisfy the governing differential flow equations whereas the domain meshes generated here are for the purpose of grouping vortons or evaluation points. The domain meshes in the present case do not have to conform to any flow boundaries, they extend only to regions where vortons or evaluation points reside, and they are generated using a very simple algorithm. Generation of adaptive domain meshes is achieved using a very small amount of CPU time whereas adaptive mesh generation for conventional CFD problems can be quite CPU time intensive.

2 SOURCE AND FIELD-POINT EXPANSIONS

As indicated previously, the basic solution method requires one to be able to write series expansions for source and field evaluation point domains. In addition, one must be able to translate the centers of the source and field point domains without actually having to evaluate the series expansions themselves. This section provides a description of the requisite expansions for the vector potential in a three-dimensional vorton field.

We begin by considering a single vorton whose vector potential $\vec{\Phi}$ is given by:

$$\vec{\Phi} = \frac{\Gamma \Delta s}{4\pi r} \vec{e}_s, \quad (1)$$

where Γ is its circulation strength, Δs is its length, r is the distance from its center to the point of evaluation, and \vec{e}_s is the unit vector in the direction of Δs or the local vorticity vector $\vec{\omega}$. The product $\Gamma \Delta s \vec{e}_s$ is another way of expressing the product of vorticity and incremental volume $\vec{\omega} \Delta v$. Equation 1 represents a core-less vorton which is singular for $r = 0$. For the purposes of generating the multipole and local series expansions r will never be equal to zero and thus, this singularity will not be encountered. A core function will be added later to accommodate the direct calculations where r may approach zero. The contribution of this vorton to the velocity field is given by:

$$\vec{U} = \nabla \times \vec{\Phi}. \quad (2)$$

2.1 Multipole Expansion About the Source Domain Center

For a cluster of vortons as shown in Figure 3 a multipole expansion can be obtained which is valid at a field point P outside of a domain D_1 enclosing the cluster. The N^{th} order expansion for the l^{th} component of $\vec{\Phi}$ is given according to [1] as:

$$\Phi_l = \sum_{n=0}^N \sum_{m=-n}^n \frac{A_n^m}{r^{n+1}} Y_n^m(\theta, \phi), \quad (3)$$

where $Y_n^m(\theta, \phi)$ is a spherical harmonic which can be defined in terms of an associated Legendre function $P_n^m(x)$ as:

$$Y_n^m(\theta, \phi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta) e^{im\phi}. \quad (4)$$

The coefficients A_n^m are given by:

$$A_n^m = \sum_{i=1}^k \frac{(\Gamma \Delta s)_{l,i}}{4\pi} \rho_i^n Y_n^{-m}(\alpha_i, \beta_i). \quad (5)$$

Here, $\rho_i, \alpha_i, \beta_i$ are the spherical coordinates of the i^{th} vorton with respect to the center of expansion. The term $(\Gamma \Delta s)_{l,i}$ is the magnitude of $\vec{\omega} \Delta v$ in the l^{th} direction of the i^{th} vorton. The first summation in Equation 3 is truncated to order N , subsequent results show that reasonably accurate results are obtained for values of N on the order of 4 or 5.

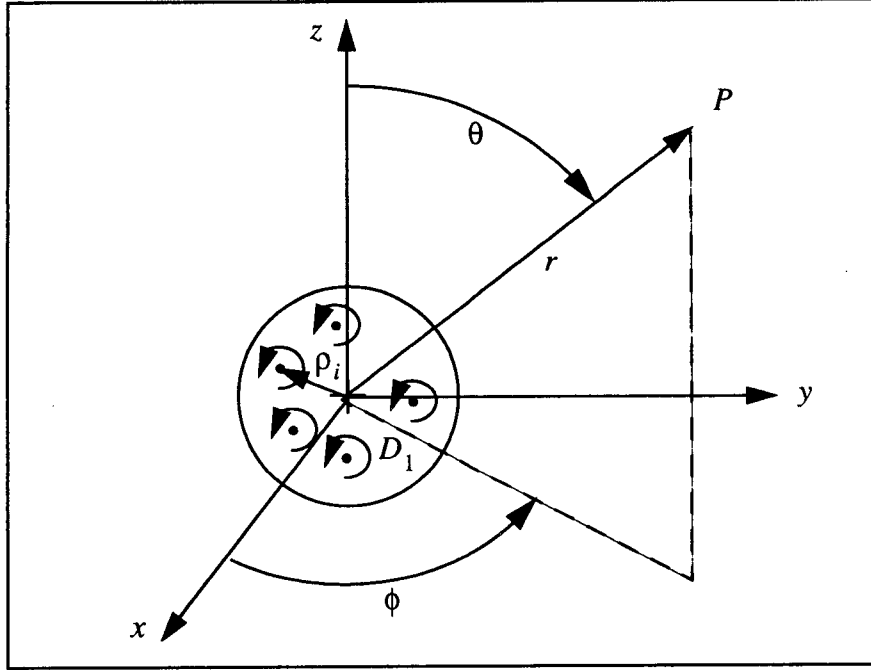


Figure 3. Multipole Expansion About Center of Source Domain

It should be noted that Greengard's [1] definition for the spherical harmonic given by Equation 4 is not the standard definition. The standard definition [5] differs from Equation 4 in that the absolute value of the index $|m|$ in Equation 4 is replaced by m and Equation 4 is multiplied by the factor $\sqrt{(2n+1)/4\pi}$. Greengard's definition is entirely appropriate for the present problem but requires consistent use in place of the standard definition. It should also be noted that only the real part of the right hand side of Equation 3 is used in computing Φ_l .

Calculations of the associated Legendre functions are obtained from the following stable recurrence relationship [6]:

$$(l-m) P_l^m = x(2l-1) P_{l-1}^m - (l+m-1) P_{l-2}^m. \quad (6)$$

Starting values required for Equation 6 for general l are given by:

$$P_m^m = (-1)^m (2m-1)!! \left(1-x^2\right)^{m/2} \quad (7)$$

$$P_{m+1}^m = x(2m+1) P_m^m.$$

Where the notation $n!!$ denotes the product of all odd integers less than or equal to n .

2.2 Translation of the Source Domain Center

As indicated previously, it is advantageous to be able to move the center of the expansion given by Equation 3 such that the coefficients of the multipole expansion in domain D_1 can be added to the coefficients of other domains contained in a larger domain D_2 . This situation is depicted in Figure 4. From reference [1] the multipole expansion for domain D_1 can be expanded about the center of domain D_2 according to:

$$\Phi_l = \sum_{j=0}^N \sum_{k=-j}^j \frac{\tilde{A}_j^k}{r^{j+1}} Y_j^k(\theta, \phi), \quad (8)$$

where the coordinates of point P with respect to the center of domain D_2 are (r, θ, ϕ) and the coefficients \tilde{A}_j^k are given by:

$$\tilde{A}_j^k = \sum_{n=0}^j \sum_{m=ml}^{mh} \frac{A_{j-n}^{k-m} J a_m^{k-m} D_n^m D_{j-n}^{k-m} \rho^n Y_n^{-m}(\alpha, \beta)}{D_j^k}. \quad (9)$$

The $J a_n^m$ coefficients are given by:

$$J a_n^m = \begin{cases} (-1)^{\min(|m|, |n|)}, & \text{if } (m \cdot n) < 0 \\ 1, & \text{otherwise} \end{cases}. \quad (10)$$

The D_n^m coefficients used in Equation 9 are given by:

$$D_n^m = \frac{(-1)^n}{\sqrt{(n-m)!(n+m)!}}. \quad (11)$$

It should be noted that in the shifting operation it is undesirable to use values of A_n^m which have indices outside of the range suggested by Equation 3. This constraint can be expressed by placing the following limits on the second sum of Equation 9:

$$ml = \max(-n, k-N) \text{ and } mh = \min(n, k+N). \quad (12)$$

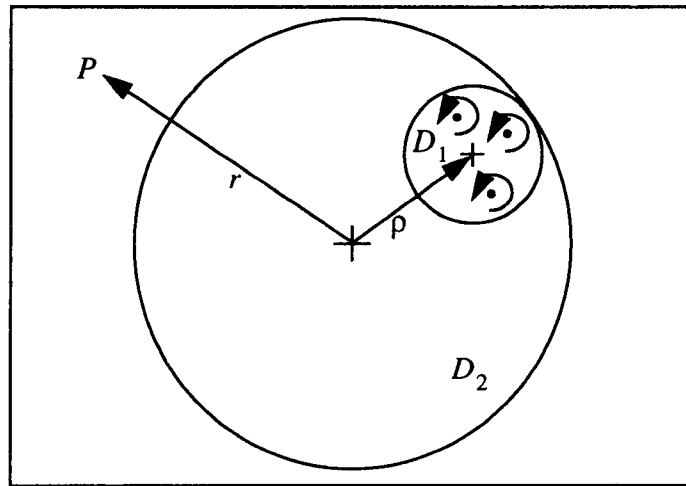


Figure 4. Translation of Source Domain Center

2.3 Local Expansion About the Field Domain Center

A local (Taylor series in r) expansion will now be obtained about the center of a field evaluation point domain D_3 due to the influence of vortices in D_2 . Referring to Figure 5, it can be seen that this expansion will be valid at field points inside D_3 if D_2 and D_3 are well separated. The local series for the l^{th} component of the vector potential at point P resulting from the expansion about the center of D_3 can be written as:

$$\Phi_l = \sum_{j=0}^N \sum_{k=-j}^j B_j^k Y_j^k(\theta, \phi) r^j, \quad (13)$$

where the coordinates of point P with respect to the center of domain D_3 are (r, θ, ϕ) and the coefficients B_j^k are given by:

$$B_j^k = \sum_{n=0}^N \sum_{m=-n}^n \frac{\tilde{A}_n^m Jab_m^{n,m} D_n^m D_j^k Y_{j+n}^{m-k}(\alpha, \beta)}{D_{j+n}^{m-k} \rho^{j+n+1}}. \quad (14)$$

The $Jab_m^{n,k}$ coefficients are given by:

$$Jab_m^{n,k} = \begin{cases} (-1)^n (-1)^{\min(|k|, |m|)}, & \text{if } (m \cdot k) > 0 \\ (-1)^n, & \text{otherwise} \end{cases}. \quad (15)$$

It should be noted that the coordinates (ρ, α, β) are the location of the center of the multipole expansion with respect to the center of the local expansion.

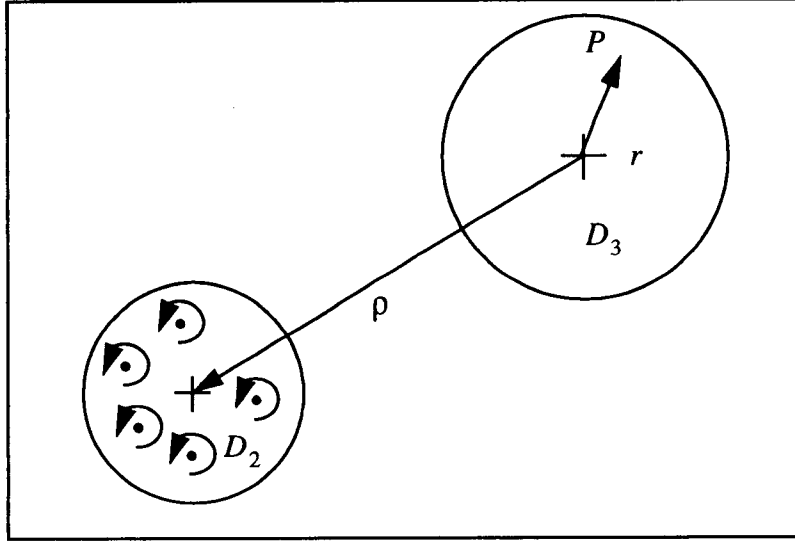


Figure 5. Local Expansion About Center of Field Domain

2.4 Translation of the Field Domain Center

The last series expansion involves the translation of the center of the field evaluation point domain. As Figure 6 depicts, the expansion for the domain D_3 can be used for any domain D_4 which is inside of D_3 by performing a simple translation of the expansion center. The

local series expansion for the l^{th} component of the vector potential at point P about the center of D_4 is:

$$\Phi_l = \sum_{j=0}^N \sum_{k=-j}^j \tilde{B}_j^k Y_j^k(\theta, \phi) r^j, \quad (16)$$

where the coordinates of point P with respect to the center of domain D_4 are (r, θ, ϕ) and the coefficients \tilde{B}_j^k are given by:

$$\tilde{B}_j^k = \sum_{n=j}^N \sum_{m=-n}^n \frac{B_n^m Jb_{n-j, m-k}^m D_{n-j}^{m-k} D_j^k Y_{n-j}^{m-k}(\alpha, \beta) \rho^{n-j}}{D_n^m}. \quad (17)$$

The $Jb_{n,m}^k$ coefficients are given by:

$$Jb_{n,m}^k = \begin{cases} (-1)^n (-1)^m, & \text{if } m \cdot k < 0 \\ (-1)^n (-1)^{k-m}, & \text{if } m \cdot k \geq 0 \text{ and } |k| < |m| \\ (-1)^n, & \text{otherwise} \end{cases} \quad (18)$$

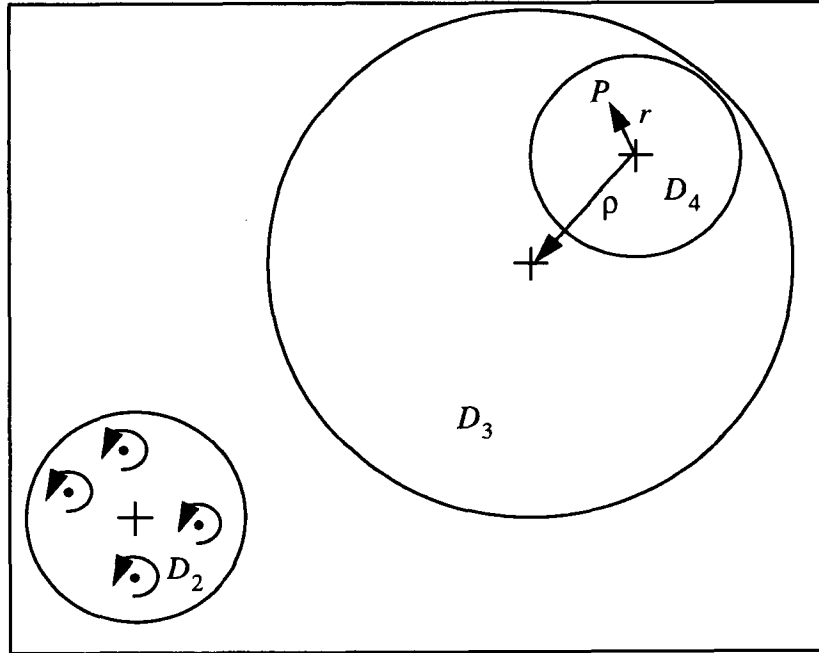


Figure 6. Translation of Field Domain Center

It should be noted that in the shifting operation that the spherical harmonic $Y_{n-j}^{m-k}(\alpha, \beta)$ is equal to zero for values of $(n-j) < |m-k|$ since the associated Legendre function $P_{n-j}^{|m-k|}(x)$ is equal to zero for those cases according to equation 8.753(3.) of reference [7]. The coordinates (ρ, α, β) are the location of the “old” center of the local expansion with respect to the “new” center of the local expansion.

2.5 Local Series Expansion Due to Single Vortons

As we shall see later, there will be a need to obtain a local series expansion for a set of vorton sources using the vorton centers in place of the center of the source domain D_2 . This occurs for situations when one cannot use the domain D_2 multipole expansion in domain D_3 since they are not well separated. This could be accomplished by formally applying the operations indicated in Equations 3 and 14 to each vorton in D_2 but would not be very efficient since only A_0 is non-zero in Equation 3. Using this fact, one can show that the appropriate values for the coefficients B_j^k for use in Equation 13 which account for the m vortons in domain D_2 can be given by:

$$B_j^k = \sum_{i=1}^m \frac{(\Gamma \Delta s)_{l,i}}{4\pi} \frac{Y_j^{-k}(\alpha_i, \beta_i)}{\rho_i^{j+1}}. \quad (19)$$

As indicated in Figure 7, the coordinates $(\rho_i, \alpha_i, \beta_i)$ are the location of the vorton centers in domain D_2 with respect to the center of the field domain D_3 .

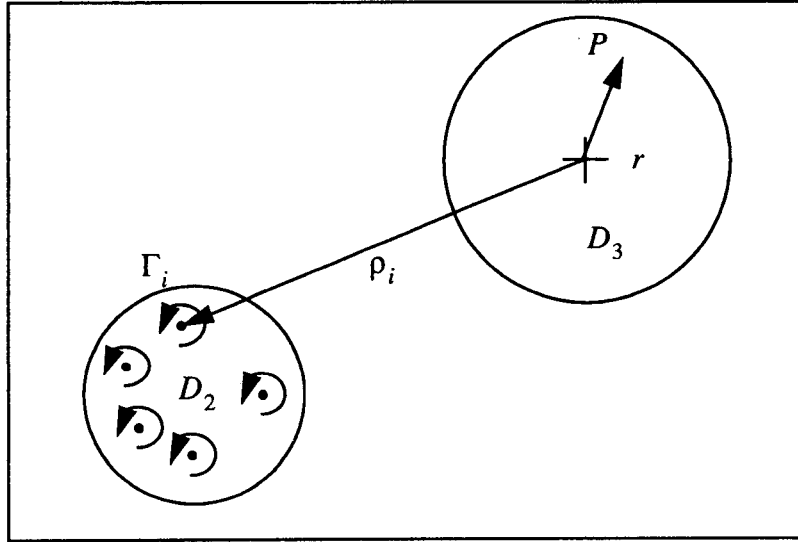


Figure 7. Local Expansion for Individual Sources

3 METHOD OF SOLUTION

The present work is implemented by the computer code FAST3D developed at Sandia National Laboratories. This code requires that one specify a distribution of vortons by their positions (in Cartesian coordinates), and the three components of $\Gamma \Delta s \underline{e}_s$ or $\vec{\omega} \Delta v$ and place such information in an input file. Positions of all field evaluation points are placed in a separate input file. The code also requires that one specify the domain of interest (the position and size of the cube which encompasses all vortons and field evaluation points, the maximum number of vortons in a box N_S , the maximum number of field evaluation points in a box N_F and the order N of the series expansions to be used. FAST3D output consists of vorton input data and the values of the three components of the vector potential and velocity at all specified field evaluation points. The following describes the methodology used in the FAST3D code.

3.1 Adaptive Mesh

As mentioned in the introduction, the spatial domains in which vortons are located are obtained in an adaptive manner which is a three-dimensional extension of the two-dimensional scheme used by Carrier, Greengard, and Rokhlin [4] for a field of charged particles. The spatial domains for the field evaluation points are obtained in an identical fashion. It should be noted that the only inherent commonality between the source and field meshes is that they both are derived from a common box (cube) which contains all of the source and field points of interest.

3.1.1 Mesh Generation

The method requires that an adaptive mesh be generated that encloses all the N_V vortons and all of the N_E evaluation points of interest in the flow. A cube with dimensions S_0 is first constructed which encloses all of the chosen vortons and evaluation points. If there happen to be vortons or evaluation points outside of the chosen box or even on the box edges, they are discarded and an appropriate warning is issued. This box is subdivided into eight equal boxes. If any of those eight boxes contain more than a specified number of vortons N_S that particular box is subdivided into eight more boxes. Vortons residing on mesh surfaces are associated with the box whose center has a more positive x , y , or z value than the x , y , or z value defining the mesh surface. This process is continued until all of the boxes contain less than the specified number of vortons. The process is repeated for the field evaluation points. Each box size is associated with a box level l with the original box level being equal to 0. Therefore at a particular level l , the box dimension S_l is given by:

$$S_l = \frac{S_0}{2^l}. \quad (20)$$

At each level, empty boxes are ignored. The ability to ignore regions of the flow-field in which vortons or evaluation points are not present is a feature of the method which provides significant time savings. Boxes which can be subdivided are parent boxes and those which cannot are childless boxes. A list of vortons or field points residing in each parent or childless box at each level is maintained. Each vorton or field point is identified by a unique label (number).

The mesh generators in the FAST3D code allows a maximum level of $l = 10$. The box dimension S_{10} at that level will be $1/1024^{th}$ of the original box dimension. In some instances there will be more than the specified number of vortons or field points in some boxes at $l = 10$. In such cases, a warning is issued and the code proceeds leaving the extra vortons or field points in the boxes. It should be pointed out that the solution resolution is not limited by the maximum level l since interactions between vortices within the smallest box are still considered. It may also be possible that a maximum number of parent or childless boxes at a given level may be exceeded. This number is currently set at 1000 in the FAST3D code. If this number is exceeded a warning is issued along with a request to increase the value of N_S or N_F .

3.1.2 Box Lists

The purpose of defining a box list for each field point evaluation box at each level is to define the separation condition between the selected field box and all source boxes in the source box mesh. This in turn allows one to efficiently calculate the influence of vortons in a particular source box on the vector potential and velocity in the field box. In general, each field box at each level has five possible relationships with each source box. The five lists define restrictions on the use of the series expansions between domains defined by the source box and field box.

In order to formalize the box list definitions, let NBOXI refer to the field box for which a box list is being developed. As mentioned previously, parent boxes are boxes which can be subdivided into eight smaller boxes, while childless boxes cannot. Thus, parent boxes contain more than N_S vortons or N_F field points. Empty boxes contain no vortons or field points, while childless boxes contain from 1 to N_S vortons or from 1 to N_F field points. A colleague box of NBOXI is a box which is adjacent to NBOXI and which has the same size (level) as NBOXI. There are at most twenty-six colleague boxes associated with NBOXI. Definitions for the five box lists are as follows:

- **Box List 1-** In order for NBOXI to have any list 1 boxes, NBOXI must itself be childless. If NBOXI is a childless box, then list 1 boxes consist of all childless source boxes at all levels which are adjacent to NBOXI and all childless source boxes which are either contained within, congruent with, or which contain NBOXI. This list defines childless source boxes which are not sufficiently separated from NBOXI to allow any of the series expansions to be used. Direct calculations must be made for this list.
- **Box List 2-** List 2 boxes of NBOXI are source boxes which occupy the same positions that would be occupied by children of the colleagues of NBOXI's parent that are well separated from NBOXI. Note that the colleagues of NBOXI's parent do not actually have to exist or to have children. NBOXI and its list 2 source boxes can be either parent or childless boxes. NBOXI and its list 2 boxes will be the same size (level) and will be separated by at least the dimension of one of their sides. For this case both source domain (multipole) and field domain (local) series expansions can be used.

- **Box List 3-** In order for NBOXI to have any list 3 boxes, NBOXI must itself be childless. List 3 boxes can be either parent or childless source boxes. List 3 boxes occupy the positions that descendants of the colleagues of NBOX would occupy. Note that the colleagues of NBOX do not actually have to exist or to have descendants. The parent of the list 3 box must be adjacent to NBOXI but the list 3 box must not itself be adjacent to NBOXI. List 3 boxes will always be smaller than NBOXI. NBOXI will be separated from the list 3 box by one box which is the same size as the list 3 box. Source domain series expansions (multipole) can be used, field domain (local) expansions cannot.
- **Box List 4-** NBOXI can be either a parent or childless box. List 4 boxes must be childless source boxes. NBOXI is in the position that a descendant of the colleagues of any list 4 box would occupy. Note that the colleagues of the list 4 box do not have to exist nor do they have to have descendants. The parent of NBOXI must be adjacent to the list 4 box but NBOXI must not itself be adjacent to the list 4 box. List 4 boxes will always be larger than NBOXI. NBOXI will be separated from the list 4 box by one box which is the same size as NBOXI. Field domain (local) series expansions can be used, source domain (multipole) expansions cannot.
- **Box List 5-** List 5 boxes consist of all source boxes that are well separated from NBOXI's parent. No calculations are necessary. Contributions from these distant boxes reside in the parent of NBOXI

In order to demonstrate how the set of box lists brings information into the field box NBOXI, consider the example given in Figures 8 and 9. These figures represent a slice through the source mesh. The field box NBOXI is shown as the cross-hatched box in Figure 8 a). Assume that the indicated mesh has been generated for a set of vortons and that there are no empty boxes. It should be pointed out that although the field box NBOXI is congruent with one of the source boxes in this example, this need not be the case in general. Since there are vortons in every source box, one would expect every box to contribute to the vector potential and velocity in NBOXI. From Figure 8 a), it can be seen that there are contributions from NBOXI's own box list and that this list contains box lists 1-4. As indicated previously, box list 5 does not contribute directly and is thus not even indicated in Figures 8 and 9. In Figure 8 b), there are several list 2 and 4 boxes that contribute to NBOXI's parent. Since this is a parent box, it does not have any list 1 or 3 boxes associated with it. The information from NBOXI's parent is fed into NBOXI by translation of the field domain center of the parent of NBOXI to the center of NBOXI. Information from NBOXI's grandparent's list boxes (see Figure 9 c)) were previously fed into NBOXI's parent in the same fashion. As indicated in Figure 9 d), NBOXI's great grandparent has a null box list and therefore does not contribute in this case. It can be noted that every box in the mesh has contributed to NBOXI. In general, it can be seen that the box list concept allows information to be passed down through successive levels in an orderly and efficient fashion.

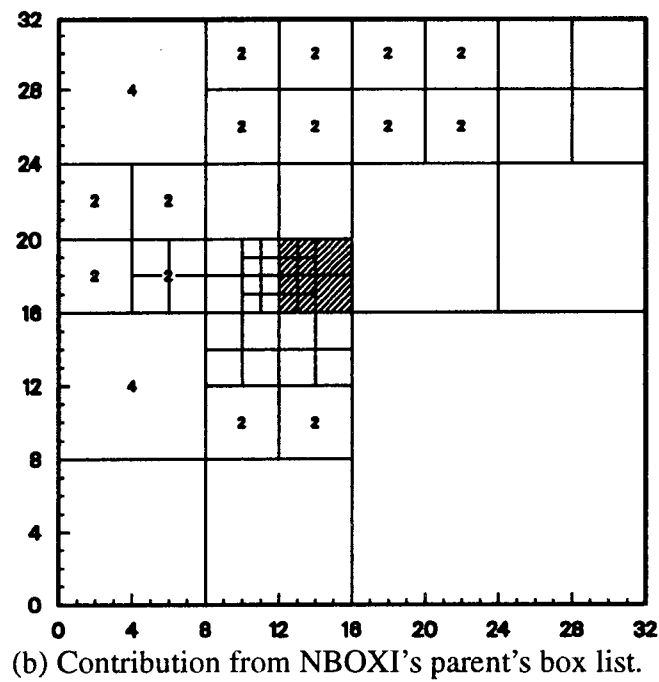
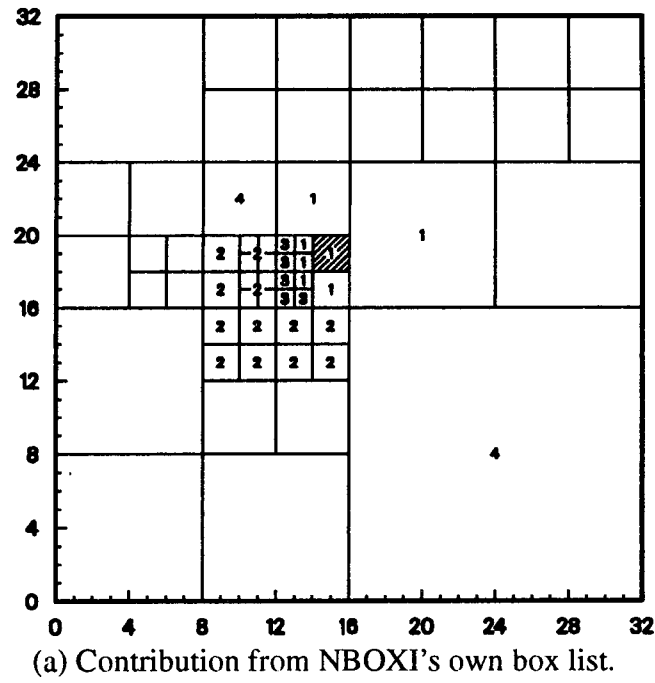
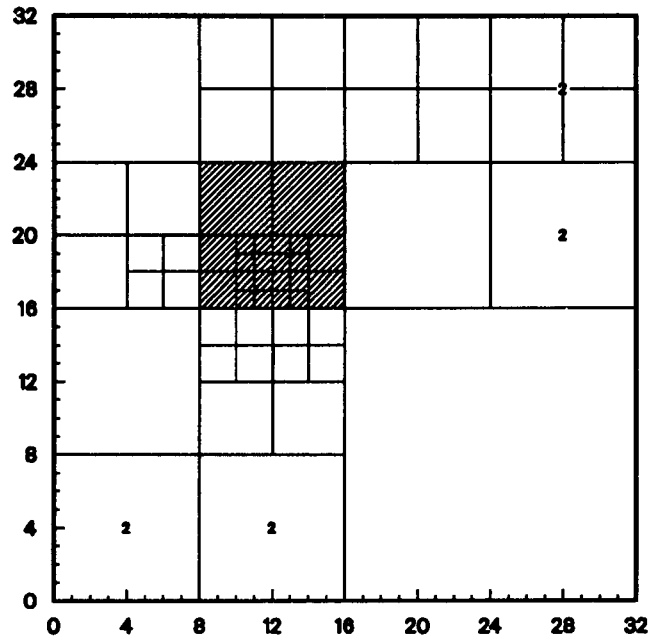
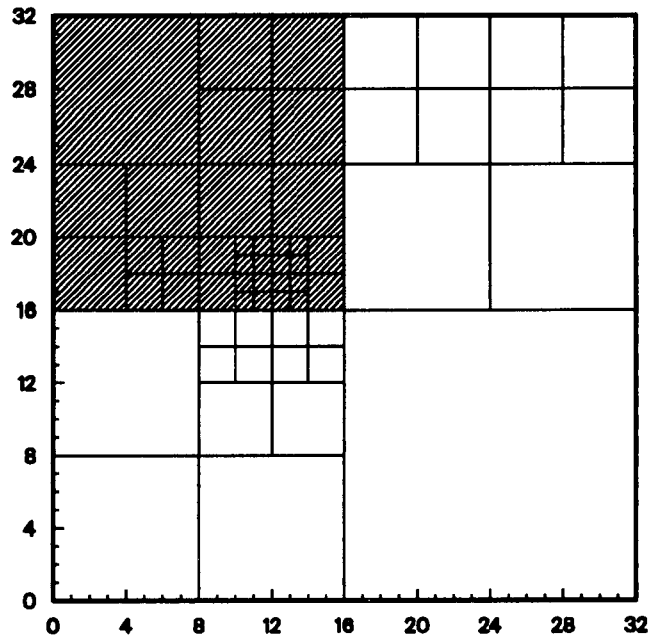


Figure 8. Box List Example



(c) Contribution from NBOXI's grandparent's box list.



(d) Contribution from NBOXI's great grandparent's box list.

Figure 9. Box List Example (continued)

3.2 Generation of Series Coefficients

As mentioned previously, the series expansions are used only in cases where the source list box associated with a field box NBOXI is a list 2, list 3, or list 4 box. Information from the multipole coefficients (the A coefficients) is used in cases where NBOXI has list 2 or list 3 boxes, while information from the local coefficients (the B coefficients) is used in cases where NBOXI has list 2 or 4 boxes. A set of A coefficients is obtained for each source box (may be either a childless or parent box) whereas a set of B coefficients is obtained for each field evaluation box (childless or parent).

3.2.1 Multipole Coefficients

The A_n^m or \tilde{A}_n^m coefficients defined in Equations 5 and 9 are used to calculate the B_n^m coefficient when the source box is a list 2 box. Also, for list 3 boxes, the A coefficients are used directly in the calculation of field variables. The A coefficients are calculated by first using Equation 5 to generate values of A_n^m for all childless source boxes. The centers of expansion for the smallest (highest level) source child boxes are then shifted to their parent's centers using Equation 9. The contributions from the child boxes to the parent box are added together producing a set of A coefficients for the parent box. This process is repeated for decreasing levels (larger boxes) until the A coefficients are calculated for all parent and childless source boxes at all levels. This process may be thought of as the process by which information concerning source domain expansions is promulgated from smaller domains into larger domains. It should be noted that there a set of A coefficients for each component of the vector potential for each source box.

3.2.2 Local Coefficients

The B coefficients are only used for list 2 and list 4 boxes. For list 2 boxes, the B_n^m coefficients can be calculated from Equation 14. For list 4 boxes, the contribution from each vortex must be considered separately since the source domain is not well separated from the center of the field box NBOXI. For this case Equation 19 is used. Therefore, the B coefficients are calculated by first using Equation 14 for list 2 boxes and Equation 19 for list 4 boxes to generate values of B_n^m for all boxes. Next, the centers of expansion for the largest (lowest level) parent boxes which have list 2 or list 4 boxes associated with them are shifted to their children's centers using Equation 17. The contribution from the parent box to the child box is added together with the list 2 and list 4 contributions associated with the child box itself. This produces a set of B coefficients for the child box. This process is repeated for increasing levels (smaller boxes) until the B coefficients are calculated for all parent and childless boxes at all levels. This process may be thought of as the process by which information concerning field domain expansions is promulgated from larger domains into smaller domains. It should be noted that there are a set of B coefficients for each component of the vector potential for each field evaluation box.

3.3 Calculation of Field Variables

The field variables $\vec{\Phi}$ and \vec{U} at a point P are calculated according to the type of list box containing the source information. For list 1 boxes, the field variable at point P must be calculated by considering the influence of each individual vorton in the source box on the point P

in the field evaluation box. For list 2 and 4 source boxes, a local series about the center of the field box is used. This series uses the B coefficients. For list 3 source boxes, a series containing the A coefficients is used.

3.3.1 Vector Potential

The vector potential at a point P in the field box for list 1 source boxes must be calculated directly. Contributions from each source in each list 1 box are added together. Each source contributes according to the following equation which is simply Equation 1 multiplied by a core function $\xi_\Phi(r/\sigma)$:

$$\vec{\Phi} = \frac{\Gamma \Delta s}{4\pi r} \xi_\Phi \vec{e}_s \quad (21)$$

Here, σ is the “core radius” associated with the vorton. The purpose of ξ_Φ is to regularize the expression for $\vec{\Phi}$ when $r = 0$ (i.e. $\vec{\Phi}$ should be finite) and to cause Equation 21 to asymptotically approach Equation 1 for r greater than a few σ . A more detailed discussion of the core function will be given in a subsequent section.

The contributions from all list 2 and 4 source boxes at a point P in a given field box are obtained by simply using the local series expansion given by Equation 13. Each field box has its own set of B coefficients which contain all of the necessary information from its list 2 and 4 source boxes as well as from list 2 and 4 boxes associated with its parent, grandparent, etc. It should be remembered that the coordinates (r, θ, ϕ) of the point P used in Equation 13 are with respect to the center of the field box.

The contributions to the vector potential from list 3 source boxes at a point P in a given field box are obtained by using the multipole expansion of Equation 3. This requires that at a point P that contributions from the multipole expansions from all list 3 source boxes be added together.

In summary, the contribution to the vector potential at a point P from list 1 and list 3 source boxes is obtained by adding the contributions from all list 1 and list 3 boxes together. The contribution from all list 2 and 4 boxes is inherent in the B coefficients which are associated with the field box itself. This contribution is added to the contributions from all list 1 and list 3 source boxes. This may be illustrated by the following equation:

$$\vec{\Phi} = \sum_{list1} \vec{\Phi}_1 + \sum_{list3} \vec{\Phi}_3 + \vec{\Phi}_{24} \quad , \quad (22)$$

where the subscripts (1,2,3,4) indicate the source box list number.

3.3.2 Velocity

The velocity at a point P is calculated using box list information in a manner similar to that for the vector potential. For example, the contribution to the velocity at a point P from list 1 and list 3 source boxes is obtained by adding the contributions from all list 1 and list 3 boxes together. The contribution from all list 2 and 4 boxes is inherent in the B coefficients which

are associated with the field box itself. This contribution is added to the contributions from all list 1 and list 3 source boxes yielding the following equation:

$$\vec{U} = \sum_{list1} \vec{U}_1 + \sum_{list3} \vec{U}_3 + \vec{U}_{24} , \quad (23)$$

where the subscripts (1,2,3,4) indicate the source box list number.

The relationship between the vector potential and the velocity vector is given by Equation 2 which can be written as:

$$\vec{U} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ \Phi_x & \Phi_y & \Phi_z \end{vmatrix} . \quad (24)$$

For list 1 source boxes it is convenient to write Equation 2 in component form in terms of the r partial derivatives of the components of the vector potential:

$$U_i = \left(\frac{x_j}{r} \right) \frac{\partial \Phi_k}{\partial r} - \left(\frac{x_k}{r} \right) \frac{\partial \Phi_j}{\partial r} . \quad (25)$$

Since the individual potential components are radially symmetric, partial derivatives with respect to θ and ϕ are zero. It should be noted that the indices (i, j, k) represent the (x, y, z) , (y, z, x) , or (z, x, y) coordinate axes. Using Equation 21 in Equation 25 results in the following expression for the i^{th} component of the velocity vector for list 1 calculations:

$$U_i = \frac{\xi_U}{4\pi r^3} [x_k (\Gamma \Delta s)_j - x_j (\Gamma \Delta s)_k] , \quad (26)$$

where

$$\xi_U \equiv \xi_\Phi - \left(\frac{r}{\sigma} \right) \xi'_\Phi . \quad (27)$$

Equation 27 provides the relationship between the velocity core function and the vector potential core function. The velocity core function $\xi_U(r/\sigma)$ will have the properties that it will cause the velocity to be equal to zero for $r/\sigma = 0$ and will approach unity for r greater than several σ .

We next address the problem of obtaining the velocity vector for situations where the A and B coefficients for list 2, 3, or 4 source boxes are available. Representation of the sources contained in these boxes is in terms of multipole or local series expansions which are not radially symmetric. This means that a simple expression such as Equation 25 is not valid for this situation. In order to develop a valid expression similar to Equation 25 which includes the addi-

tional angular dependencies we obtain the partial derivatives $\partial\Phi_l/\partial x_j$ and use them in Equation 24 to calculate the velocity components. The method for doing this is to first obtain the set of partial derivatives in spherical coordinates and then transform these (using the chain rule) into cartesian coordinates. This transformation is given by:

$$\begin{Bmatrix} \frac{\partial\Phi_l}{\partial x} \\ \frac{\partial\Phi_l}{\partial y} \\ \frac{\partial\Phi_l}{\partial z} \end{Bmatrix} = \begin{bmatrix} \sin\theta\cos\phi, & \frac{\cos\theta\cos\phi}{r}, & -\frac{\sin\phi}{r\sin\theta} \\ \sin\theta\sin\phi, & \frac{\cos\theta\sin\phi}{r}, & \frac{\cos\phi}{r\sin\theta} \\ \cos\theta, & \frac{-\sin\theta}{r}, & 0 \end{bmatrix} \begin{Bmatrix} \frac{\partial\Phi_l}{\partial r} \\ \frac{\partial\Phi_l}{\partial\theta} \\ \frac{\partial\Phi_l}{\partial\phi} \end{Bmatrix}. \quad (28)$$

The vector on the right hand side of Equation 28, is obtained by symbolically differentiating the multipole and local series expansions. The partial derivative $\partial\Phi_l/\partial r$ for the multipole expansion is obtained from differentiation of Equation 3 resulting in:

$$\frac{\partial\Phi_l}{\partial r} = - \sum_{n=0}^N \sum_{m=-n}^n (n+1) \frac{A_n^m}{r^{n+2}} Y_n^m(\theta, \phi). \quad (29)$$

For the local expansion, Equation 13 is differentiated, resulting in:

$$\frac{\partial\Phi_l}{\partial r} = \sum_{j=0}^N \sum_{k=-j}^j j B_j^k Y_j^k(\theta, \phi) r^{j-1}. \quad (30)$$

Partial derivatives with respect to θ and ϕ require one to differentiate $Y_n^m(\theta, \phi)$. The differentiation with respect to ϕ is the easiest and is given by:

$$\frac{\partial}{\partial\phi} \left(Y_n^m(\theta, \phi) \right) = im Y_n^m(\theta, \phi). \quad (31)$$

Using Equations 3 and 13, the partial derivative $\partial\Phi_l/\partial\phi$ for the multipole and local series expansions is given by:

$$\frac{\partial\Phi_l}{\partial\phi} = \sum_{n=0}^N \sum_{m=-n}^n im \frac{A_n^m}{r^{n+1}} Y_n^m(\theta, \phi), \quad (32)$$

$$\frac{\partial\Phi_l}{\partial\phi} = \sum_{j=0}^N \sum_{k=-j}^j im B_j^k Y_j^k(\theta, \phi) r^j.$$

respectively, where i is the imaginary number $\sqrt{-1}$.

Upon examination of Equation 4, it can be seen that differentiation of $Y_n^m(\theta, \phi)$ with respect to θ , requires one to obtain the derivative of $P_n^{|m|}(\cos\theta)$. Defining a new variable $\xi \equiv \cos\theta$ allows one to write:

$$\frac{\partial}{\partial\theta} P_n^{|m|}(\xi) = \frac{\partial}{\partial\xi} P_n^{|m|}(\xi) \frac{\partial\xi}{\partial\theta} = -\sin\theta \frac{\partial}{\partial\xi} P_n^{|m|}(\xi) . \quad (33)$$

There are several possible differentiation formulas for the function $P_n^{|m|}(\xi)$. The following formula obtained from equation 8.733(1.) of reference [7] was chosen for its eventual ease of use in the series expansions:

$$\sin^2(\theta) \frac{\partial}{\partial\xi} P_n^{|m|}(\xi) = -\sin\theta P_n^{|m|+1}(\xi) - |m| \cos\theta P_n^{|m|}(\xi) . \quad (34)$$

Combining Equations 33 and 34 yields:

$$\frac{\partial}{\partial\theta} P_n^{|m|}(\xi) = P_n^{|m|+1}(\xi) + |m| \frac{\cos\theta}{\sin\theta} P_n^{|m|}(\xi) . \quad (35)$$

The derivative of $Y_n^m(\theta, \phi)$ with respect to θ is then given by:

$$\frac{\partial}{\partial\theta} \left(Y_n^m(\theta, \phi) \right) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \left(P_n^{|m|+1}(\xi) + |m| \frac{\cos\theta}{\sin\theta} P_n^{|m|}(\xi) \right) e^{im\phi} . \quad (36)$$

Using Equations 3 and 13, the partial derivative $\partial\Phi_l/\partial\theta$ for the multipole and local series expansions is given by:

$$\frac{\partial\Phi_l}{\partial\theta} = \sum_{n=0}^N \sum_{m=-n}^n \frac{A_n^m}{r^{n+1}} \frac{\partial}{\partial\theta} \left(Y_n^m(\theta, \phi) \right) , \quad (37)$$

$$\frac{\partial\Phi_l}{\partial\theta} = \sum_{j=0}^N \sum_{k=-j}^j B_j^k \frac{\partial}{\partial\theta} \left(Y_j^k(\theta, \phi) \right) r^j .$$

It should be noted that the transformation matrix in Equation 28 is undefined for $r = 0$ and for $\sin\theta = 0$. Therefore, each of these occurrences must be treated as special cases. The possibility that $r = 0$ occurs only for the local expansion. For that case, the derivatives $\partial\Phi_l/\partial x_j$ can be written directly as:

$$\begin{aligned} \frac{\partial\Phi_l}{\partial x} &= -\frac{1}{\sqrt{2}} \left(B_1^1 + B_1^{-1} \right) , \\ \frac{\partial\Phi_l}{\partial y} &= -\frac{i}{\sqrt{2}} \left(B_1^1 - B_1^{-1} \right) , \\ \frac{\partial\Phi_l}{\partial z} &= B_1^0 . \end{aligned} \quad (38)$$

It has been verified that these limits are approached in a smooth fashion as one allows r to approach zero. The case where $\sin\theta = 0$ cannot be treated quite so cleanly. This case was treated by never allowing $\sin\theta$ to be exactly zero. The following scheme was used which forces $|\sin\theta|$ to be greater than zero by forcing θ to be greater than 0 but less than 180 :

$$\theta = \theta + \epsilon \cos\theta \text{ for } |\sin\theta| < \epsilon . \quad (39)$$

A value of 0.001 was arbitrarily chosen for ϵ which dictates a range of θ values between 0.057° and 179.943° .

In summary, the contribution to the velocity vector at a point P from a source in a list 1 box is obtained using Equation 26. The contributions to the velocity vector at a point P from all list 2 and 4 source boxes are obtained by first calculating the partial derivatives $\partial\Phi_l/\partial x_j$ from local series expansions using Equations 30, 32, and 37 or Equation 38 in Equation 28. These partial derivatives are then inserted into Equation 24. The contributions to the velocity vector from list 3 source boxes at a point P in a given field box are obtained by first calculating the partial derivatives $\partial\Phi_l/\partial x_j$ from multipole series expansions using Equations 29, 32, and 37 in Equation 28. These partial derivatives are then inserted into Equation 24.

4 CORE FUNCTION AND SELF INDUCTION

4.1 Core Functions

A complete discussion of the suitability of core functions is beyond the scope of the present work. An excellent presentation of such matters is given by Winckelmans and Leonard [8]. In the present work, we choose to use a Gaussian smoothing scheme. One of the reasons for choosing this scheme is that both ξ_U and ξ_Φ are equal to unity after $\rho \equiv r/\sigma$ increases above a relatively small value ($\rho > 1.75$). Other smoothing schemes tend to have values of ξ_U and ξ_Φ which approach unity at somewhat greater values of ρ . For Gaussian smoothing, the equation for ξ_U is given by:

$$\xi_U = 1 - e^{-\rho^3} . \quad (40)$$

In order to obtain ξ_Φ , Equation 27 is first rewritten as:

$$\frac{\xi_U}{\rho^2} = - \frac{d}{d\rho} \left(\frac{\xi_\Phi}{\rho} \right) \quad (41)$$

or

$$U^+ = - \frac{d\Phi^+}{d\rho} . \quad (42)$$

where $U^+ \equiv \xi_U/\rho^2$ and $\Phi^+ \equiv \xi_\Phi/\rho$. Solving Equation 42 for Φ^+ yields:

$$\Phi^+ = \Phi_0^+ - \int_0^\rho U^+ d\rho , \quad (43)$$

where $U^+ = (1 - e^{-\rho^3})/\rho^2$. Since Φ^+ must be equal to zero for $\rho = \infty$ then;

$$\Phi_0^+ = \int_0^\infty \frac{(1 - e^{-\rho^3})}{\rho^2} d\rho . \quad (44)$$

Numerical integration of Equation 44 yields a value of $\Phi_0^+ = 1.354$. Equation 43 is next integrated numerically with the results shown in Figure 10 as the “exact” curve. The function ξ_Φ was then obtained from $\xi_\Phi = \rho \Phi^+$ and is shown in Figure 11 as the “exact” curve. The numerically obtained representation of ξ_Φ was then fitted with the following curve:

$$\xi_\Phi = 1 - e^{-q\rho} , \quad (45)$$

where

$$q = 1.354 + 0.842\rho + 0.559\rho^2 . \quad (46)$$

The fitted data for Φ^+ and ξ_Φ are shown in Figures 10 and 11 respectively.

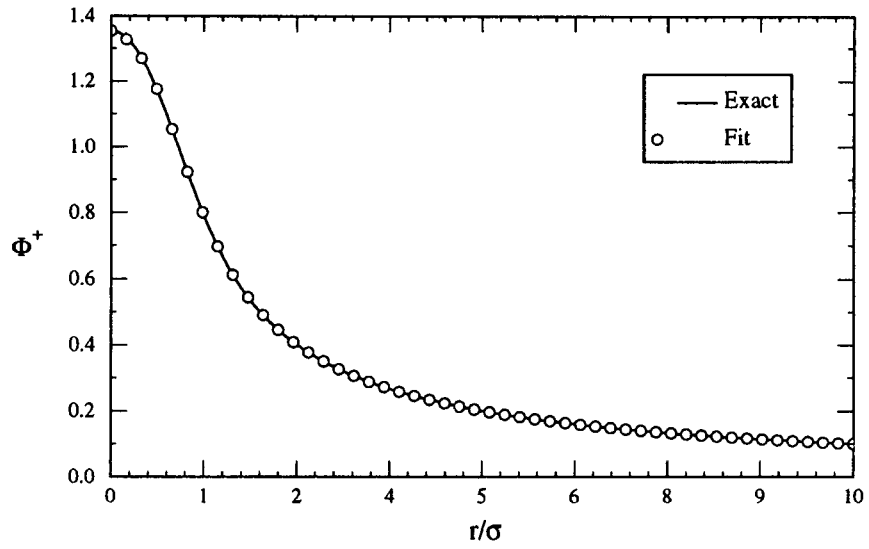


Figure 10. Potential Function for Gaussian Smoothing

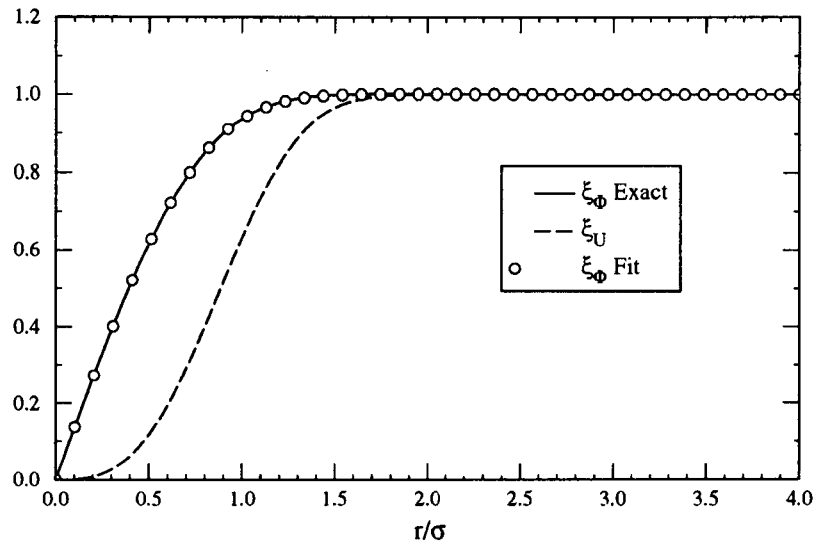


Figure 11. Gaussian Core Functions

4.2 Self Induction

4.2.1 Velocity

Due to the nature of the core function, the velocity of a vorton at its center due to self induction is zero as calculated by the FAST3D code. Self induction must be added by the user as is appropriate. For instance, in order to properly simulate the self induction of a vortex ring by using vortons alone, the cores must overlap by at least one core radius. However, if one uses a small number of large vortons then the vortex ring may possess an artificially thick core which will cause the ring to move at the wrong speed in the fluid. To overcome this, one may use a large number of vortons for cases where the core radius to ring radius is small. An alternative solution is to use cores whose sizes are based upon the physics of the flow but which may not overlap and then apply a local correction term.

One such local correction scheme is based upon Kempka's [9] "Local Induction Approximation" which provides the self induced velocity of a small segment of a ring whose arc length is Δs and whose local radius of curvature is R . This correction uses a modified low order algebraic core function which is given by:

$$\xi_U = \frac{\rho^3}{(\rho^2 + \alpha)^{3/2}}. \quad (47)$$

The modification consists of replacing the number 1 in the denominator of the low order algebraic core function as given by Winckelmans and Leonard [8] with α . The constant α is equal to 0.413 for Gaussian cores and 0.223 for constant vorticity cores. This core function gives rise to the following equation for the magnitude of the self induced velocity U_s :

$$U_s = \frac{\Gamma R}{8\pi} \int_{-\frac{\Delta s}{2}}^{\frac{\Delta s}{2}} \frac{s^2}{(s^2 + \alpha\sigma^2)^{3/2}} ds \quad (48)$$

or

$$\frac{U_s R}{\Gamma} = -\frac{1}{4\pi} \left[\frac{1}{\beta} + \frac{1}{2} \ln \left(\frac{\beta - 1}{\beta + 1} \right) \right], \quad (49)$$

where β is defined as:

$$\beta \equiv \sqrt{1 + \alpha \left(\frac{2\sigma}{\Delta s} \right)^2}. \quad (50)$$

An example problem is shown in Figure 12 for a circular ring which is simulated by 40 vortons. The solid line represents the desired solution and is obtained from an equation due to Strickland and Amos [2] for circular rings with constant vorticity cores ($\alpha = 0.223$). The diamond shaped symbols represent the uncorrected data using 40 vortons while the circular symbols represent the corrected data. As can be seen from this figure, the correction given by Equation 49 works very well.

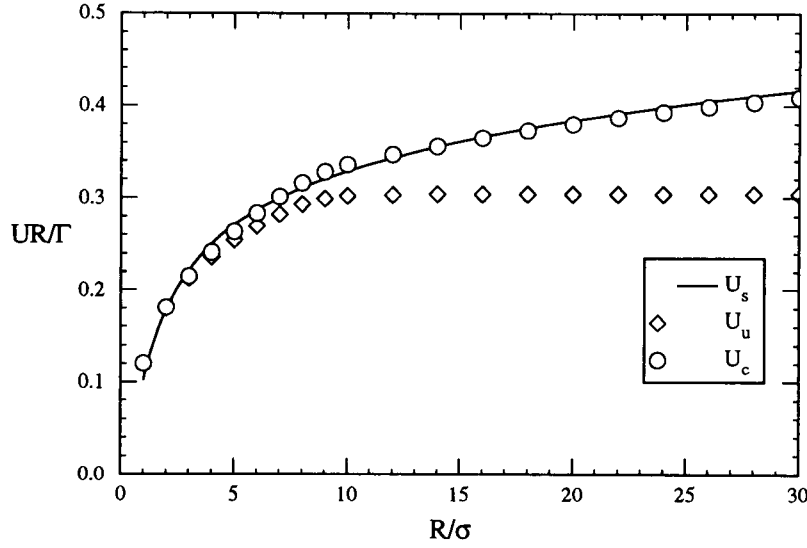


Figure 12. Self Induction of a Circular Ring

4.2.2 Potential

Due to the nature of the core function, the potential of a vorton at its center is not zero as calculated by the FAST3D code. For instance, when Gaussian smoothing is used, the value of Φ^+ at $\rho = 0$ is equal to 1.354 as indicated in Figure 10. This value is not, however, indicative of the integrated value from segments making up the element which the vorton represents. In general, it is too high especially if the cores do not overlap. The local potential must therefore be adjusted by the user as is appropriate. A local correction can be developed by integrating the potential over the element in question. For instance, for a vortex filament where $\Delta s/R$ is small, the self induced potential Φ_s on the arc of length Δs is given by:

$$\Phi_s^+ \equiv \frac{4\pi\Phi_s\sigma}{\Gamma\Delta s} = \frac{2\sigma}{\Delta s} \int_0^{\frac{\Delta s}{2\sigma}} \Phi^+ d\left(\frac{r}{\sigma}\right). \quad (51)$$

If one uses the modified low order core function then:

$$\Phi^+ = \left(\rho^2 + \alpha\right)^{-\frac{1}{2}} \quad (52)$$

and

$$\Phi_s^+ = \frac{2\sigma}{\Delta s} \ln \left[\frac{\frac{\Delta s}{2\sigma} + \sqrt{\left(\frac{\Delta s}{2\sigma}\right)^2 + \alpha}}{\sqrt{\alpha}} \right]. \quad (53)$$

The original computation for the self induced potential may be corrected by subtracting 1.354 from Equation 53 and applying the correction to the original result. Figure 13 illustrates the error associated with a ring comprised of 40 vortons whose directions are tangent to the ring. The “exact” solution is obtained by placing 2000 vortons on the ring while using the Gaussian smoothing. The “corrected” solution is obtained using $\alpha = 0.413$ in Equation 53. As can be seen from this figure, the corrected potential is reasonably well predicted while the potential which uses a self induced value of 1.354 is about 30% too high for a ring radius to core radius of 30. For 20 vortons the error increases to about 90% while the corrected calculation is virtually the same as for 40 vortons.

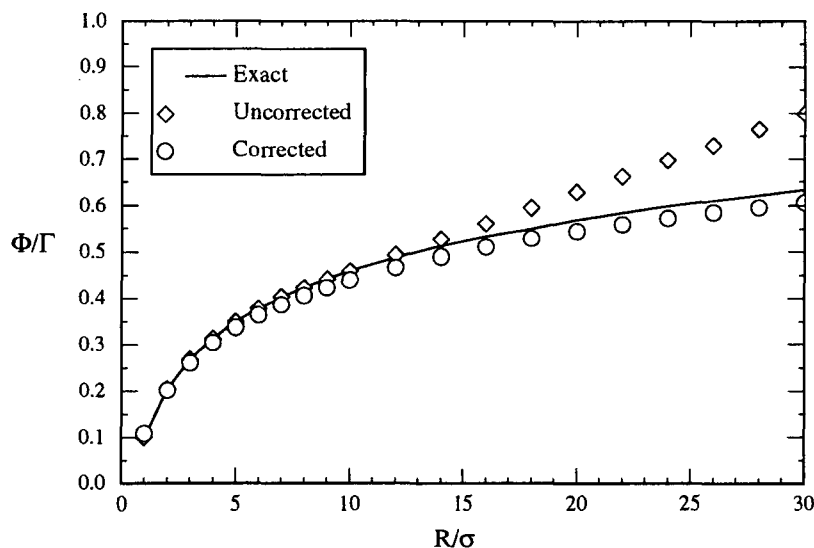


Figure 13. Potential on a Circular Ring

Several remarks will be made in closing this section. First of all, if one desires only the velocity field and does not need values of the potential then the local corrections for the potential are unnecessary. Calculation of the velocity field using FAST3D is not dependent on these local corrections. Secondly, it should be noted that a somewhat more complicated correction algorithm has been developed using Gaussian smoothing in place of the modified low order algebraic core function. Results of this correction are only slightly better than those shown in Figure 13. The reason for this is that the correction is not very sensitive to the exact core function used since values of $\Delta s/\sigma$ tend to be large where the correction is important (i.e., large R/σ). The value of Φ^+ for ρ greater than about 2 or 3 is independent of the core function and thus the integral expressed in Equation 51 is somewhat invariant to the core function for cases where $\Delta s/\sigma$ tends to be large.

5 BENCHMARK TESTS

5.1 General Description of Benchmark Tests

The important dependent variables in the bench mark tests are the CPU run times and the truncation errors associated with the calculation of field variables. The independent variables are the number of vorton sources and evaluation points in the field, the distribution of the sources and evaluation points in the field, the number of terms used in the multipole and local series expansions, and the maximum number of source and evaluation points allowed in any childless box in the source and field meshes respectively. In view of the large number of parametric variations which might ultimately be useful, we have run a somewhat limited set of test cases to examine truncation error and CPU run times. Three general configurations of vorton and field evaluation point placements have been studied as indicated in Figure 14. We lack motivation at this point in time to be more exhaustive since we plan to enhance the algorithm during the next fiscal year which should reduce the CPU times significantly. The enhancements will consist of adding Greengard's and Rokhlin's FFT shift algorithm [3] and a change in the procedure to calculate field variables produced by list 3 source boxes.

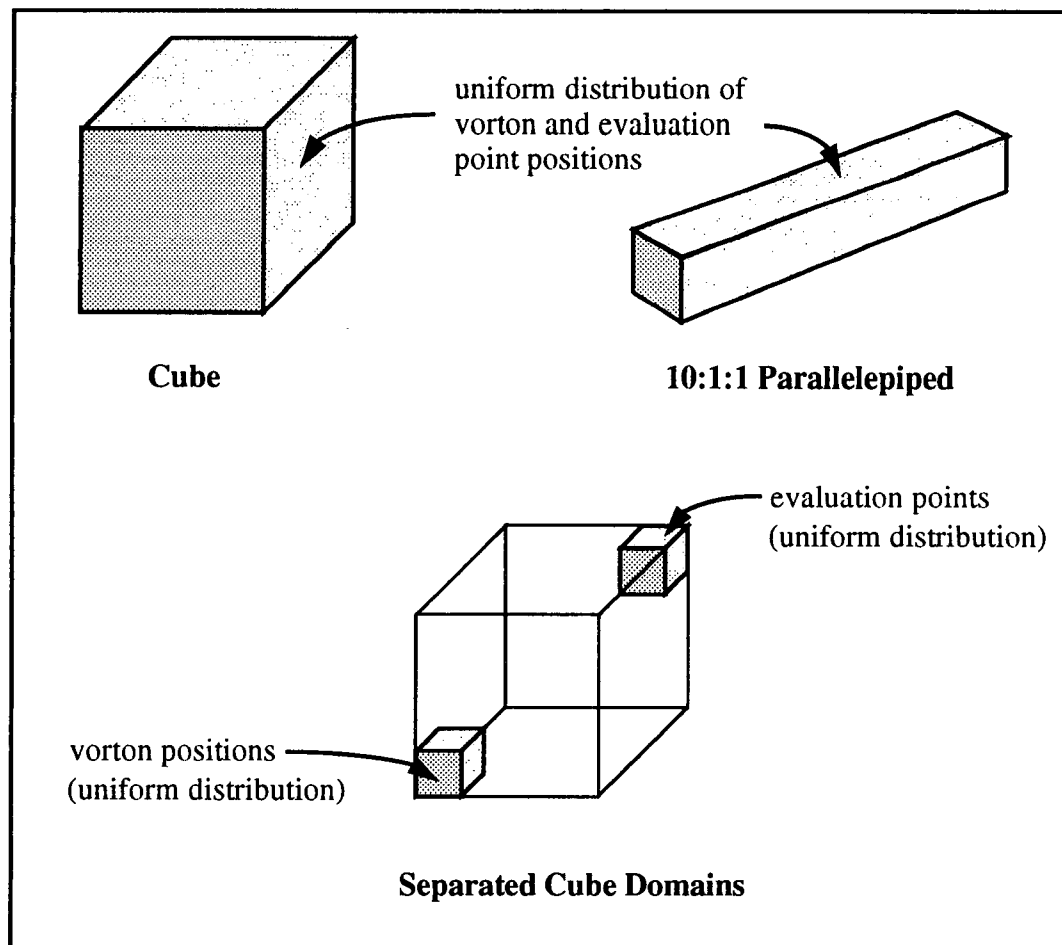


Figure 14. Benchmark Configurations

Before presenting the results for the various benchmark cases, a metric for the truncation error must be defined. One such measure of the error incurred by truncation of the multipole and local series after the N^{th} order terms can be obtained for the magnitude of the vector potential and velocity vector fields by means of the following equations:

$$\epsilon_{\Phi} = \left[\frac{\sum_{i=1}^{N_v} (|\Phi|_i - |\Phi|_{ei})^2}{\sum_{i=1}^{N_v} |\Phi|_{ei}^2} \right]^{\frac{1}{2}},$$

(54)

$$\epsilon_U = \left[\frac{\sum_{i=1}^{N_v} (|U|_i - |U|_{ei})^2}{\sum_{i=1}^{N_v} |U|_{ei}^2} \right]^{\frac{1}{2}}.$$

Here, $|\Phi|_i$ and $|U|_i$ are the magnitudes of the vector potential and velocity vector at point i as calculated from the truncated series. The quantities $|\Phi|_{ei}$ and $|U|_{ei}$ are the magnitudes of the vector potential and velocity vector at point i as calculated exactly using the direct method. As indicated, these error calculations are obtained from summations over all of the N_v vortons in the flow. Assuming that single precision accuracy means that any number stored in the computer will be accurate through 6 significant digits, then single precision round off errors will produce values of ϵ_{Φ} and ϵ_U up to about 1×10^{-5} . Therefore, errors calculated from Equation 54 which are equal to or less than 1×10^{-5} imply that the machine round off errors are possibly as large as the truncation errors. For such cases, one should not continue to add terms to the series expansions for added accuracy unless the machine precision is also increased.

5.2 Results From Cubic Configuration

Truncation and accuracy data obtained from a number of cubic configuration simulations are given in Appendix A.1. In each case, a uniform distribution of vortons was placed in a cube with sides equal to 4 with one corner at the origin and another at (4,4,4). The vortons along the box border were shifted toward the inside of the box by a distance of 0.02 to avoid their rejection by the mesh generator. Evaluation points were also placed at the same points. Truncation errors, as defined by Equation 54, for the magnitude of the vector potential and the velocity vector fields are plotted in Figure 15 for a field of 9261 vortons. This allows one to gain some insight into the magnitude of such errors. These errors are plotted versus N which is the order of the last term in the multipole or local series expansion. From Figure 15 it can be noted that the truncation error is independent of the maximum number of vortons N_{vc} in a cell or box. For values of N equal to 4 and 5, the errors associated with the vector potential are on the order of 2×10^{-5} and 4×10^{-6} respectively. Thus, truncation errors for the vector potential with N equal to 4 or 5 are of the same order of magnitude as possible machine

round off errors and are therefore acceptable for single precision accuracy. Truncation errors associated with the velocity vector for N equal to 4 and 5 are approximately equal to 8×10^{-4} and 1×10^{-4} respectively. For the magnitude of the velocity vector, one must decide whether or not errors of such magnitude are acceptable since they are one or two orders of magnitude larger than machine round off errors. One would suspect that, in most cases, velocity calculations that are accurate to within 0.01 % to 0.08 % will be acceptable. Deciding whether one should use N equal 4 or 5 is also a trade off between truncation error and CPU time t_{cpu} . In order to gain some appreciation for the difference in t_{cpu} for $N = 4$ versus $N = 5$, consider the plot shown in Figure 16. As can be seen from this figure, t_{cpu} increases by roughly 60% between $N = 4$ and $N = 5$.

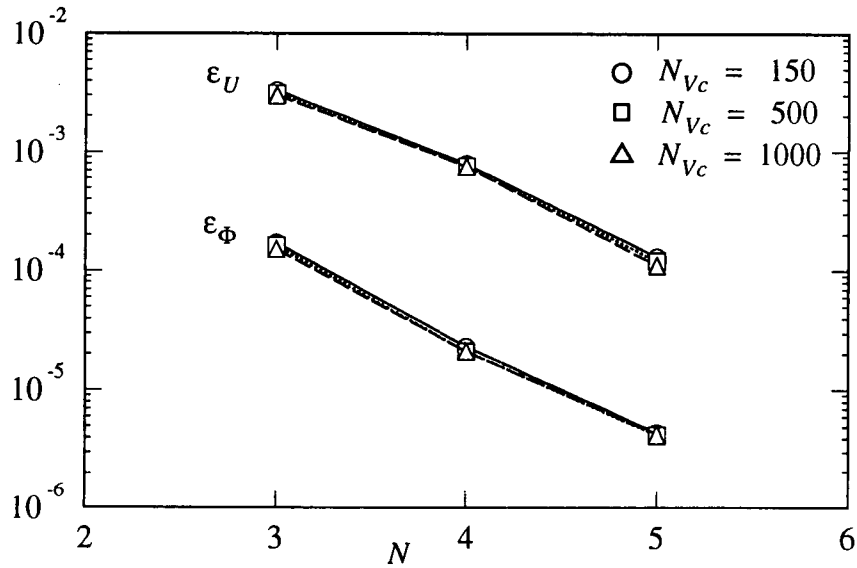


Figure 15. Error Plots for Uniform Distribution of 9261 Vortons in a Cube

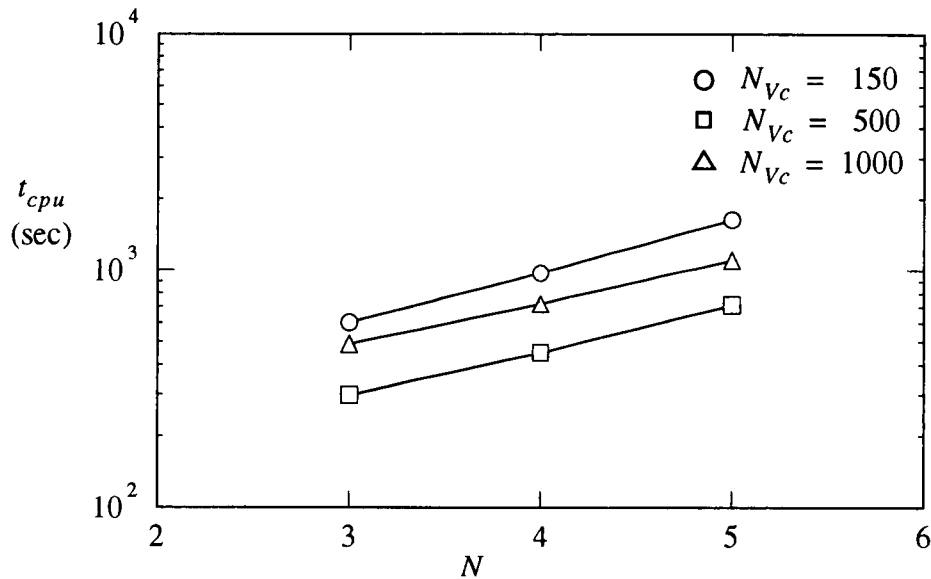


Figure 16. CPU Times for Uniform Distribution of 9261 Vortons in a Cube

The next task is to optimize the maximum number of vortons in a box. For instance, for the 9261 vorton case, the optimal value is somewhere between 200 and 900 vortons as evidenced by Figure 17. The optimal range of values for N_{vc} coincides roughly with a division of the large cube of dimension 4 into 64 smaller cubes. This would indicate a range for N_{vc} of between $N_v/64$ (145) and $N_v/8$ (1158). It should be noted that the optimal range is independent of the order N . For the $N_v = 21952$ case, the optimal range for N_{vc} with N equal to both 4 and 5 is between 400 and 2700. For the $N_v = 91125$ case, the large cube of dimension 4 is optimally divided into 512 smaller cubes. Therefore, the optimal range of N_{vc} is from 178 to 1424. From this limited set of data, the optimal range for N_{vc} is between 400 and 900. A value of 750 was arbitrarily chosen from this range as an optimal value for N_{vc} . This value is considerably higher than those used in two-dimensional planar [4] and axisymmetric fast solvers [2] where N_{vc} is equal to 20 to 30. The fact that the geometry is three-dimensional would boost the expected optimal values to only 90 to 164. The high optimal value of N_{vc} is at least due in part to the relatively expensive cost of shifting the series expansions as discussed by Greengard and Rokhlin [3]. Their FFT shift algorithm applied to the present work will most certainly lower the optimal value of N_{vc} and produce smaller CPU times.

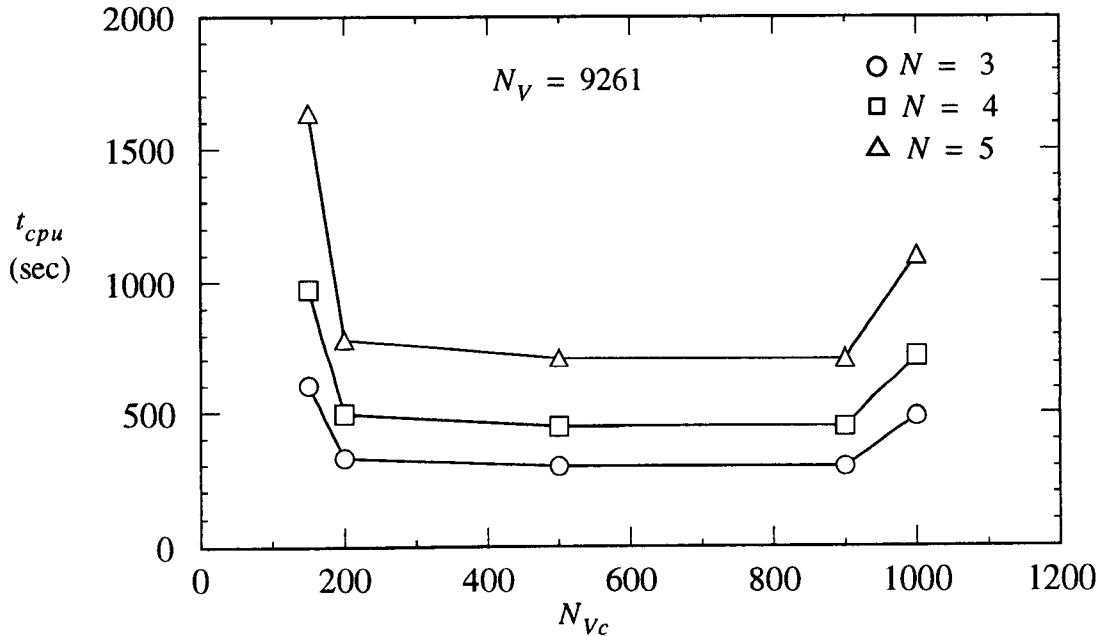


Figure 17. CPU Time Versus Maximum Number of Vortons in a Box

In summary, if we assume that we may use an expansion with order equal to $N = 4$ and a value of $N_{vc} = 750$ then the CPU time versus the number of vortons N_v in the cube is given in Figure 18. The runs were all executed on a SUN Sparc10 workstation. The CPU time for the fast solver for the cubic configuration may be approximated by:

$$t_{cpu} = 7.97 \times 10^{-3} N_v^{1.21} . \quad (55)$$

The truncation errors for the vector potential and velocity vector are given in Figure 19 as a function of the number of vortons N_v in the cube. As can be seen from Figure 19, the order

of magnitude of the truncation errors are relatively independent of the number of vortons for N_V greater than 6000 (8x750) which is the demarcation between a direct solution and a fast solution. For $N_V < 6000$ the 4x4x4 cube is divided into only 8 smaller cubes which makes the 8 source boxes list 1 boxes relative to the 8 field evaluation boxes. Thus the calculation should yield single precision accuracy which it does.

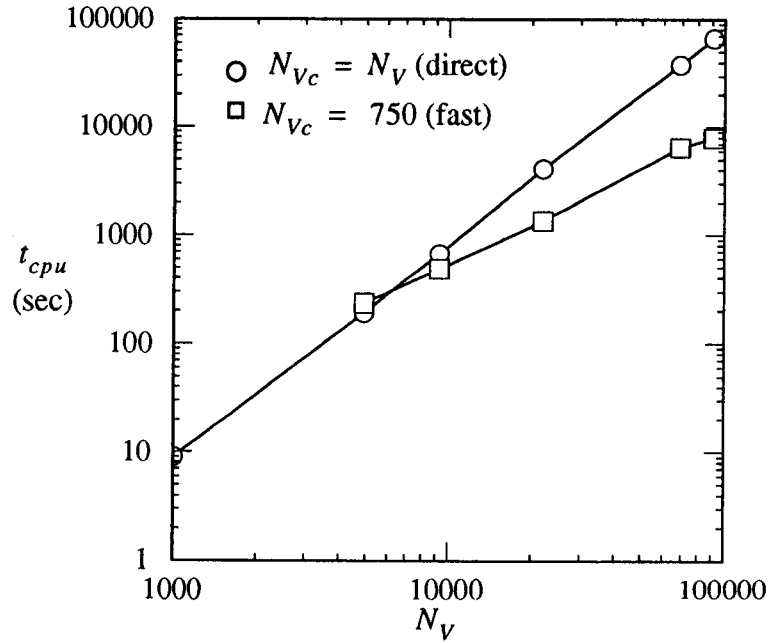


Figure 18. CPU Time For Uniform Distribution in a Cube

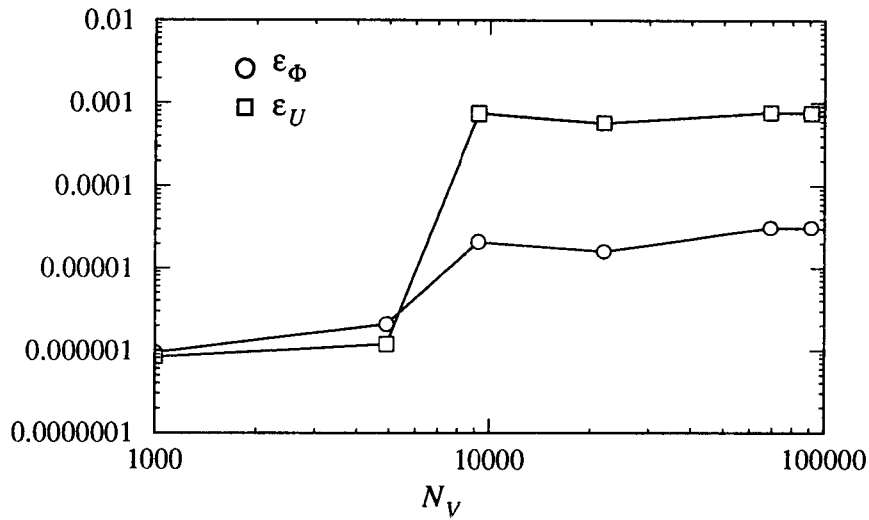


Figure 19. Truncation Errors For Uniform Distribution in a Cube

5.3 Results From 10:1:1 Parallelepiped Configuration

Truncation and accuracy data obtained from a number of parallelepiped configuration simulations are given in Appendix A.2. In each case, a uniform distribution of vortons was placed in a parallelepiped with sides equal to 4.0, 0.4 and 0.4 with one corner at the origin and another at (4.0,0.4,0.4). The vortons along the box border were shifted toward the inside of the box by a distance of 0.02 along the x axis and by 0.002 along the y and z axis to avoid their rejection by the mesh generator. Evaluation points were also placed at the same points. Examination of the data in Appendix A.2 reveals that in most cases, the optimization for N_{vc} from the cubic configuration is also valid for the parallelepiped configuration. Using a value of $N_{vc} = 750$ and a value of $N = 4$, the resulting CPU time as a function of the number of vortons in the field as shown in Figure 20 may be approximated by:

$$t_{cpu} = 1.99 \times 10^{-3} N_V^{1.32}. \quad (56)$$

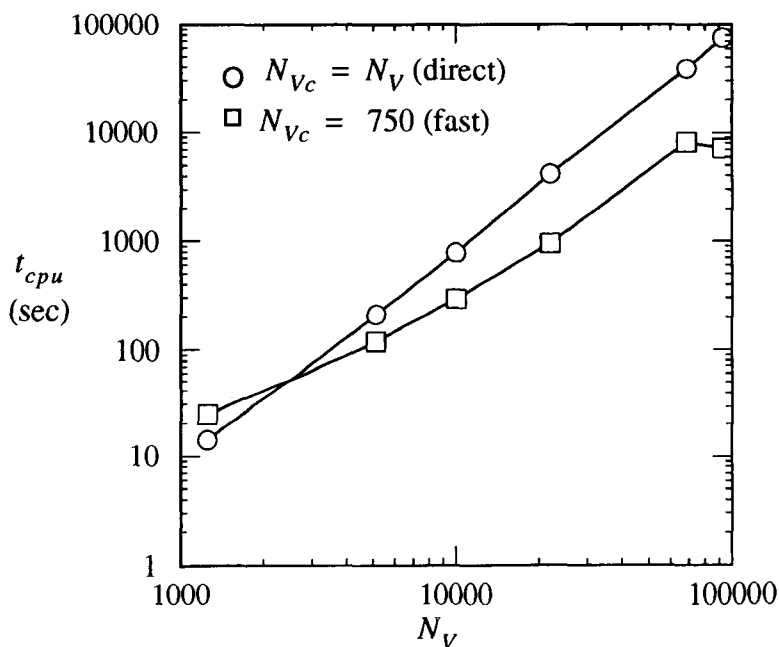


Figure 20. CPU Time For Uniform Distribution in a Parallelepiped

A comparison of Figures 18 and 20 reveals that the “break-even point” is lower for the parallelepiped configuration (i.e. the fast solver is faster than the direct solver with $N_V > 2500$ for the parallelepiped and $N_V > 6000$ for the cube). This occurs since the parallelepiped is broken up by the meshing process into well-separated domains at a lower value of N_V . For instance, for N_V slightly in excess of 1500, the parallelepiped is broken into 4 domains and the cube into 8 domains. None of the 8 domains of the cube are well separated from each other and thus all calculations are direct. Each of the 4 domains of the parallelepiped, on the other hand, are well separated from 1 or 2 of the other domains thus allowing a more efficient calculation to be made. At $N_V = 100,000$ the CPU times are very nearly the same for both configurations. The differences in Equations 55 and 56 are perhaps worth noting. The con-

stant coefficients differ by factor of 4 while the exponents differ by about 8%. It is unclear what the CPU time trends for the two configurations might be for $N_V > 100,000$ especially since an extrapolation of Equations 55 and 56 predicts a crossing at $N_V \approx 300,000$. One suspects that the two curves will merge, hopefully displaying the exponent of 1.21 instead of 1.32.

The truncation errors for the parallelepiped are shown in Figure 21. The errors are seen to be independent of N_V over most of the range. Single precision errors occur until N_V increases above 1500 at which point the parallelepiped is divided into 4 domains and the fast solver begins to work. These truncation errors are somewhat higher than for the cube configuration but are probably still acceptable for most engineering calculations.

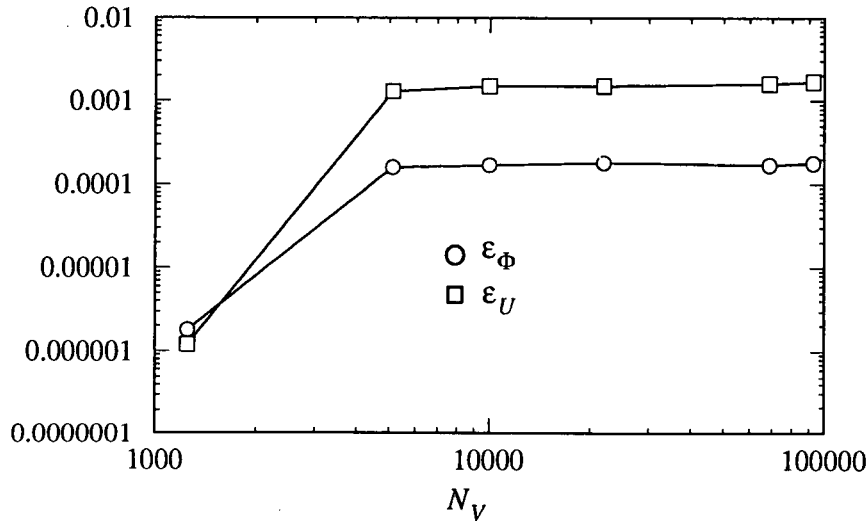


Figure 21. Truncation Errors For Uniform Distribution in a Parallelepiped

5.4 Results From Separated Cube Domain Configuration

Truncation and accuracy data obtained from a number of “separated cube domain” configuration simulations are given in Appendix A.3. In each case, a uniform distribution of vortons was placed in a cube with sides equal to 1.0 with one corner at the origin and another at (1,1,1). The vortons along the box border were shifted toward the inside of the box by a distance of 0.005 to avoid their rejection by the mesh generator. A uniform set of evaluation points was placed in a separate cube with sides equal to 1.0 with one corner at (3,3,3) and another at (4,4,4). The evaluation points along the box border were also shifted toward the inside of the box by a distance of 0.005. For all of the cases presented herein, the number of evaluation points is the same as the number of vortons. The computational box used by the FAST3D code for this case is a cube with sides equal to 4 with one corner located at the origin and the other at (4,4,4).

The most efficient calculation for this geometry would require one to not subdivide either the of the two small boxes which contains the vortons and the evaluation points. However, since the present FAST3D code creates domain boundaries by requiring that less than some speci-

fied number of points reside in any childless box, then some value somewhat less than N_V but greater than $N_V/8$ is used for N_{Vc} . This causes the two cubes to each be subdivided into 8 smaller cubes. If we had used $N_{Vc} \geq N_V$ then both the vortons and the field evaluation points would have resided in congruent (4x4x4) boxes which would have made the vorton source box a list 1 box of the field evaluation box forcing direct calculations to be made. It should be noted that only a very small time penalty is paid for the subdivision into 8 source boxes and 8 evaluation boxes. One might also add an alternate input feature to the FAST3D code in which the source and evaluation domains are meshed by specifying a fixed level with respect to the global domain (i.e. the 4x4x4 cube in this case). Thus, a level 2 mesh applied to the 4x4x4 global domain would isolate the 1x1x1 vorton box and the 1x1x1 evaluation box without subdividing either of them.

The CPU time for the direct and fast calculation for this case is shown in Figure 22. The CPU time for the fast solution technique is approximated by:

$$t_{cpu} = 8.66 \times 10^{-3} N_V^{1.00}. \quad (57)$$

As indicated by Equation 57, the increase in CPU time for the fast solver is linear for this case. The “break-even” point is at about 1000 vortons and the fast solver is seen to be an additional order of magnitude faster than the direct solver with each order of magnitude increase in the number of vortons. Thus for $N_V = 100,000$ the fast solver will obtain solutions in about 1% of the time required by the direct solution technique.

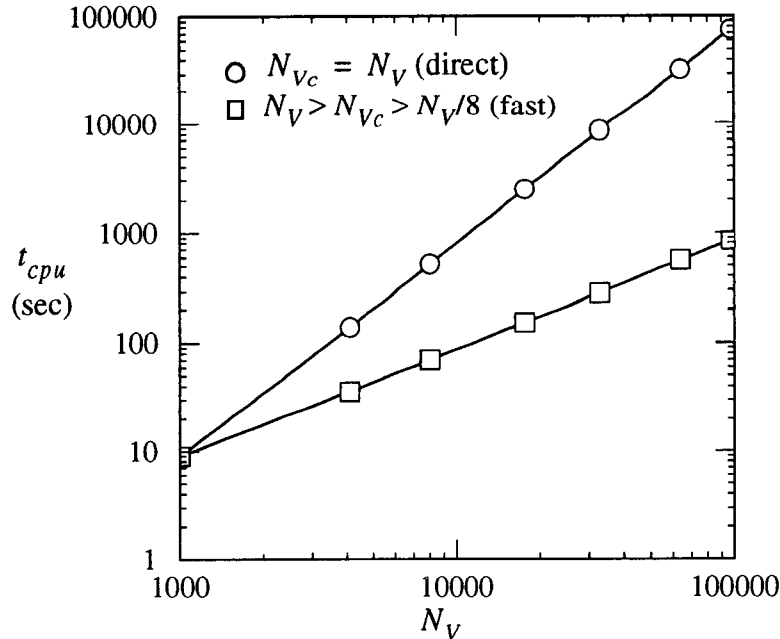


Figure 22. CPU Time For Uniform Distribution in a Separated Cube Domain

The truncation errors for the separated cube domain are shown in Figure 23. The errors are seen to be independent of N_V over the entire range. The truncation errors for the velocity calculations are somewhat higher than for the cube configuration and slightly higher than for the parallelepiped. The velocity errors are still probably acceptable for most engineering calculations. The truncation errors for the vector potential, on the other hand, are considerably lower than those for the parallelepiped configuration and slightly smaller than for the cube configuration. One explanation for such behavior is that the vector potential is almost constant across the evaluation domain which reduces its truncation errors. On the other hand, the velocity vector tends to be small which may increase its truncation error. It should be noted that the resultant direction of the vector potential for each of the vortons is parallel to the diagonal of the $4 \times 4 \times 4$ global domain box which connects the $1 \times 1 \times 1$ vorton source domain and the $1 \times 1 \times 1$ field evaluation domain. The resultant velocity field in the evaluation domain is one which displays a weak rotation about this connecting diagonal.

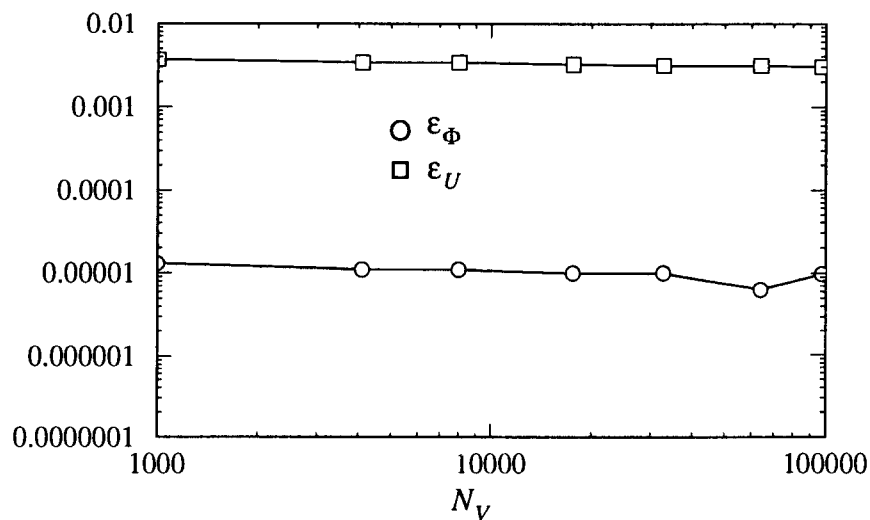


Figure 23. Truncation Errors For Uniform Distribution in a Separated Cube Domain

5.5 Lessons Learned from Benchmark Test Results

From the three sets of benchmark tests it is apparent that the CPU times are very dependent upon the configuration. A major factor is the fraction of source domains that are well separated from the evaluation domains. For example, the CPU time was lowest for the last set of benchmark tests (the “separated cube domain”) in which 100% of the source and evaluation domains were well separated. On the other hand, the poorest CPU performance came from the first benchmark test in which the fraction of well-separated source and evaluation domains tends to be the lowest of the three configurations studied. Consider for example, an interior evaluation box in the uniform cube configuration. The interior box has 27 neighboring source boxes that are not well separated. If there are 64 source boxes, then interactions from 42% of the source boxes must be computed directly. As the number of boxes in the domain increases, the fraction of direct calculations decreases which is why the CPU times for the cube configuration and the parallelepiped configuration tend to approach each other for larger values of N_V (larger N_V implies larger numbers of boxes for optimum results).

From the three sets of benchmark tests it is seen that the accuracy of the velocity vector is not greatly affected by the configuration, always being on the order of 1×10^{-3} for a value of $N = 4$. For the same precision, the accuracy of the vector potential is on the order of 1×10^{-5} for the cube and separated cube configurations and on the order of 1×10^{-4} for the parallelepiped configuration. One explanation for this behavior is that the vector potential for the parallelepiped is more variable across the evaluation domain which increases its truncation errors since the higher order terms in the series tend to die out less rapidly. The accuracy is no doubt affected by the orientation and relative magnitudes of the vortons as well as their placement in the source domains.

There are at least two time-saving features which should be incorporated into the present FAST3D code prior to its use as a production fast solver. The first of these requires the use of Greengard's and Rokhlin's FFT method [3] to speed up the operations associated with shifting the multipole and local series expansions to new centers. These shift operations currently require $O(N^4)$ CPU time whereas the FFT method requires $O(N^2 \log N)$ CPU time. For $N = 4$ it is estimated that the calculations for the uniform cube configuration would be 2 to 3 times faster. Greengard and Rokhlin found that using the FFT's on the same configuration with $N = 8$ was 8 times faster. The optimum maximum number of vortons in a box will no doubt decrease with a corresponding increase in the number boxes. The second feature is simply a rearrangement of the procedure for calculating field variables, especially for calculating the contribution of list 1 and list 3 boxes. The present arrangement for searching for list 1 and list 3 boxes becomes very inefficient for situations where there are a large number of source boxes. Presently, the searches are conducted while calculating contributions to each evaluation point. The new method will require searches to be made for each evaluation box not each evaluation point. The expected impact of this change will be to allow the code to optimize with a larger number of source boxes and thus allow the potential gains from the FFT shift algorithm to be realized.

The last general observation which can be made regarding the benchmark tests is that values of N_S and N_F , which are the maximum number of vortons in a box and maximum number of evaluation points in a box respectively, need to be optimized. It is clear from these limited benchmark tests that this optimization is configuration dependent especially when one begins to consider source and evaluation configurations which are not congruent. Optimization algorithms need to be developed which consider the configuration aspects of the problem. One may find that there are more appropriate criteria for subdividing a domain other than N_S and N_F . It should be pointed out that any optimization scheme must be able to deal with the large quantum jumps associated with subdividing a cube into 8 smaller cubes. For example, with the present scheme, a box which is slightly overpopulated will be divided into 8 boxes, several of which, if not all of which, will be grossly underpopulated.

6 SUMMARY

The work accomplished can be summarized as follows:

- A Fortran computer code FAST3D has been written to generate solutions for the vector potential and velocity vector fields for an arbitrary three-dimensional distribution of vortons and an arbitrary three-dimensional distribution of evaluation points. Multipole and local series expansions written in terms of spherical harmonics were used to make calculations for the influence of groups of vortons on well-separated groups of evaluation points. Simple adaptive domain decomposition algorithms were used which dismiss regions of the global domain in which no vortons or evaluation points are present. Labeling methodology associated with source box lists was extended to be able to handle the non-congruent nature of the vorton and evaluation point fields.
- Preliminary benchmark tests were run on three basic configurations of the vorton and evaluation point field. A total of 195 runs were made in which the CPU time as well truncation errors were obtained. For multipole and local series expansions of order $N = 4$ the error associated with the velocity vector was on the order of 10^{-3} whereas the error associated with the vector potential was on the order of 10^{-4} or better. No improvement in CPU time was obtained for less than 1000 vortons. For 100,000 vortons, the CPU time using the fast solver was from 1% to 10% of that required by the direct solution technique.

Suggestions for future work can be summarized as follows:

- Two time-saving features should be incorporated into the present FAST3D code prior to its use as a production fast solver. The first of these requires the use of Greengard's and Rokhlin's FFT method [3] to speed up the operations associated with shifting the multipole and local series expansions to new centers. If one uses multipole and local series expansions of order $N = 4$ it is estimated that the calculations for the uniform cube configuration would be 2 to 3 times faster. The second feature requires that the procedure for calculating field variables be modified. The present arrangement for searching for list 1 and list 3 boxes becomes very inefficient for situations where there are a large number of source boxes. The new method will require searches to be made for each evaluation box instead for each evaluation point as is presently done. The expected impact of this change will be to allow the potential gains from the FFT shift algorithm to be realized.
- More advanced dynamic algorithms need to be developed to better optimize values of N_S and N_F which are the maximum number of vortons in a box and maximum number of evaluation points in a box respectively. Optimization algorithms need to be developed which consider the configuration aspects of the problem. One may find that there are more appropriate criteria for subdividing a domain other than N_S and N_F . In addition, any optimization scheme must be able to deal with the large quantum jumps associated with subdividing a cube into 8 smaller cubes. For example, with the present scheme, a box which is slightly overpopulated will be divided into 8 boxes, several of which, if not all of which, will be grossly underpopulated.

7 REFERENCES

- [1] Greengard, Leslie, "The Rapid Evaluation of Potential Fields in Particle Systems," Yale University Report YALEU/DCS/RR-533, April 1987.
- [2] Strickland, J. H. and Amos, D. E., "A Fast Solver for Systems of Axisymmetric Ring Vortices," Sandia National Laboratory Report SAND90-1925, 52 pages, Also *AIAA Journal*, Vol.30, No.3, pp. 737-746, March 1992.
- [3] Greengard, L. and Rokhlin, V., "On the Efficient Implementation of the Fast Multipole Algorithm," Yale University Report YALEU/DCS/RR-602, February 1988.
- [4] Carrier, J., Greengard, L., and Rokhlin, V., "A Fast Adaptive Multipole Algorithm for Particle Simulations," *SIAM Journal on Scientific and Statistical Computing*, Vol.9, No.4, pp.669-696, July 1988.
- [5] Jackson, John D., *Classical Electrodynamics*, Second Edition, John Wiley and Sons, p. 99, 1975.
- [6] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes*, Second Edition, Cambridge University Press, Cambridge, p.247, 1992.
- [7] Gradshteyn, I. S. and Ryzhik, I. M., *Table of Integrals, Series, and Products*, Academic Press, p. 1008, 1980.
- [8] Winckelmans, G. S. and Leonard, A., "Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows," *Journal of Computational Physics*, No.109, pp.247-273, 1993.
- [9] Kempka, S. N., "The Evolution of Vortices in a Turbulent Boundary Layer," Ph.D. Dissertation, University of Illinois, pp. 217-218, (1988)

APPENDIX

A.1 Data for Cubic Configuration

Table 1: Cubic Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
1000	1000	1	9	0.0×10^{-0}	0.0×10^{-0}
1000	100	4	212	9.6×10^{-4}	1.6×10^{-5}
1000	100	3	87	3.7×10^{-3}	1.1×10^{-4}
1000	200	4	17	1.0×10^{-6}	2.5×10^{-6}
1000	200	5	21	1.0×10^{-6}	2.5×10^{-6}
1000	150	5	21	1.0×10^{-6}	2.5×10^{-6}
1000	125	5	21	1.0×10^{-6}	2.5×10^{-6}
1000	200	5	21	1.0×10^{-6}	2.5×10^{-6}
1000	200	3	14	1.0×10^{-6}	2.5×10^{-6}
1000	125	3	14	1.0×10^{-6}	2.5×10^{-6}
1000	250	3	14	1.0×10^{-6}	2.5×10^{-6}
1000	450	3	14	1.0×10^{-6}	2.5×10^{-6}
1000	999	5	21	1.0×10^{-6}	2.5×10^{-6}
1000	100	5	452	1.2×10^{-4}	4.6×10^{-6}
4913	4913	1	191	0.0×10^{-0}	0.0×10^{-0}
4913	250	4	293	2.3×10^{-3}	2.3×10^{-5}
4913	700	4	869	5.3×10^{-4}	7.5×10^{-6}
4913	350	4	293	2.3×10^{-3}	2.3×10^{-5}
4913	450	4	294	2.3×10^{-3}	2.3×10^{-5}
4913	550	4	830	2.2×10^{-3}	2.2×10^{-5}
4913	150	4	293	2.3×10^{-3}	2.3×10^{-5}
4913	900	4	234	1.2×10^{-6}	2.1×10^{-6}
4913	800	4	234	1.2×10^{-6}	2.1×10^{-6}
9261	9261	1	670	0.0×10^{-0}	0.0×10^{-0}

Table 1: Cubic Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
9261	500	4	448	7.5×10^{-4}	2.1×10^{-5}
9261	200	4	494	7.5×10^{-4}	2.1×10^{-5}
9261	800	4	481	7.5×10^{-4}	2.1×10^{-5}
9261	1000	4	723	7.5×10^{-4}	2.1×10^{-5}
9261	150	4	972	7.9×10^{-4}	2.3×10^{-5}
9261	900	4	449	7.5×10^{-4}	2.1×10^{-5}
9261	150	5	1635	1.3×10^{-4}	4.4×10^{-6}
9261	200	5	785	1.2×10^{-4}	4.2×10^{-6}
9261	500	5	715	1.2×10^{-4}	4.2×10^{-6}
9261	900	5	715	1.2×10^{-4}	4.2×10^{-6}
9261	1000	5	1100	1.1×10^{-4}	4.3×10^{-6}
9261	150	3	602	3.3×10^{-3}	1.7×10^{-4}
9261	200	3	327	3.1×10^{-3}	1.6×10^{-4}
9261	500	3	298	3.1×10^{-3}	1.6×10^{-4}
9261	900	3	298	3.1×10^{-3}	1.6×10^{-4}
9261	1000	3	487	3.0×10^{-3}	1.5×10^{-4}
21952	21952	1	4102	0.0×10^{-0}	0.0×10^{-0}
21952	300	3	1924	3.4×10^{-3}	1.7×10^{-4}
21952	400	3	1216	2.6×10^{-3}	1.4×10^{-4}
21952	1500	3	1216	2.6×10^{-3}	1.4×10^{-4}
21952	2700	3	1216	2.6×10^{-3}	1.4×10^{-4}
21952	2800	3	4121	2.2×10^{-6}	6.1×10^{-6}
21952	750	4	1357	5.8×10^{-4}	1.6×10^{-5}
21952	300	4	4153	7.1×10^{-4}	1.8×10^{-5}
21952	400	4	1365	5.8×10^{-4}	1.6×10^{-5}
21952	1500	4	1404	5.8×10^{-4}	1.6×10^{-5}
21952	2200	4	1385	5.8×10^{-4}	1.6×10^{-5}

Table 1: Cubic Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
21952	2700	4	1367	5.8×10^{-4}	1.6×10^{-5}
21952	2800	4	3969	2.2×10^{-6}	6.0×10^{-6}
21952	750	5	1676	8.5×10^{-5}	6.3×10^{-6}
21952	300	5	9228	1.2×10^{-4}	6.6×10^{-6}
21952	400	5	1755	8.5×10^{-5}	6.3×10^{-6}
21952	1500	5	1762	8.5×10^{-5}	6.3×10^{-6}
21952	2200	5	1676	8.5×10^{-5}	6.3×10^{-6}
21952	2700	5	1674	8.5×10^{-5}	6.3×10^{-6}
21952	2800	5	4095	2.2×10^{-6}	6.0×10^{-6}
68921	68921	1	37901	0.0×10^{-0}	0.0×10^{-0}
68921	750	4	6477	7.7×10^{-4}	3.1×10^{-5}
68921	5000	4	10268	1.2×10^{-3}	3.3×10^{-5}
91125	91125	1	66216	0.0×10^{-0}	0.0×10^{-0}
91125	750	4	7962	7.6×10^{-4}	3.1×10^{-5}

A.2 Data for 10:1:1 Parallelepiped Configuration

Table 2: Parallelepiped Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
1250	1250	1	14	0.0×10^{-0}	0.0×10^{-0}
1250	350	4	19	1.1×10^{-3}	1.5×10^{-4}
1250	160	5	97	7.9×10^{-4}	4.1×10^{-5}
1250	160	4	55	1.7×10^{-3}	1.7×10^{-4}
1250	100	5	121	8.2×10^{-4}	4.1×10^{-5}
1250	250	4	17	1.7×10^{-3}	1.7×10^{-4}
1250	100	4	62	1.8×10^{-3}	1.7×10^{-4}
1250	300	4	35	1.3×10^{-3}	1.6×10^{-4}
1250	200	4	17	1.7×10^{-3}	1.7×10^{-4}
1250	180	4	17	1.7×10^{-3}	1.7×10^{-4}
1250	170	4	53	1.7×10^{-3}	1.7×10^{-4}
1250	280	4	17	1.7×10^{-3}	1.7×10^{-4}
1250	290	4	17	1.7×10^{-3}	1.7×10^{-4}
1250	750	4	24	1.2×10^{-6}	1.8×10^{-6}
1250	170	5	97	7.9×10^{-4}	4.1×10^{-5}
1250	180	5	25	7.9×10^{-4}	4.1×10^{-5}
1250	250	5	25	7.9×10^{-4}	4.1×10^{-5}
1250	290	5	25	7.9×10^{-4}	4.1×10^{-5}
1250	300	5	58	5.0×10^{-4}	3.7×10^{-5}
1250	310	5	58	5.0×10^{-4}	3.7×10^{-5}
1250	320	5	58	5.0×10^{-4}	3.7×10^{-5}
1250	330	5	25	4.3×10^{-4}	3.5×10^{-5}
1250	350	5	25	4.3×10^{-4}	3.5×10^{-5}
1250	320	4	35	1.3×10^{-3}	1.6×10^{-4}
1250	360	4	19	1.1×10^{-3}	1.5×10^{-4}
5120	5120	1	208	0.0×10^{-0}	0.0×10^{-0}

Table 2: Parallelepiped Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ε_U	ε_Φ
5120	250	4	136	1.5×10^{-3}	1.6×10^{-4}
5120	250	5	212	6.7×10^{-4}	4.0×10^{-5}
5120	350	4	136	1.5×10^{-3}	1.6×10^{-4}
5120	400	4	136	1.5×10^{-3}	1.6×10^{-4}
5120	500	4	136	1.5×10^{-3}	1.6×10^{-4}
5120	600	4	136	1.5×10^{-3}	1.6×10^{-4}
5120	700	4	118	1.3×10^{-3}	1.6×10^{-4}
5120	900	4	118	1.3×10^{-3}	1.6×10^{-4}
5120	1200	4	119	1.3×10^{-3}	1.6×10^{-4}
5120	1300	4	174	8.5×10^{-4}	1.4×10^{-4}
5120	1500	4	174	8.5×10^{-4}	1.4×10^{-4}
5120	2000	4	194	8.5×10^{-4}	1.4×10^{-4}
5120	2600	4	252	1.3×10^{-6}	1.3×10^{-6}
5120	150	4	136	1.5×10^{-3}	1.6×10^{-4}
5120	100	4	1724	1.6×10^{-3}	1.6×10^{-4}
5120	150	5	212	6.7×10^{-4}	4.0×10^{-5}
5120	600	5	211	6.7×10^{-4}	4.0×10^{-5}
5120	700	5	142	6.7×10^{-4}	4.0×10^{-5}
5120	1200	5	143	6.2×10^{-4}	3.9×10^{-5}
5120	1300	5	197	3.2×10^{-4}	3.2×10^{-5}
5120	1500	5	196	3.2×10^{-4}	3.2×10^{-5}
5120	2000	5	199	3.2×10^{-4}	3.2×10^{-5}
5120	2600	5	270	1.3×10^{-6}	1.3×10^{-6}
10000	10000	1	779	0.0×10^{-0}	0.0×10^{-0}
10000	250	4	497	1.5×10^{-3}	1.7×10^{-4}
10000	500	4	291	1.5×10^{-3}	1.7×10^{-4}
10000	750	4	293	1.5×10^{-3}	1.7×10^{-4}

Table 2: Parallelepiped Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
10000	1000	4	291	1.5×10^{-3}	1.7×10^{-4}
10000	1400	4	362	1.3×10^{-3}	1.7×10^{-4}
10000	1250	4	532	1.3×10^{-3}	1.6×10^{-4}
10000	200	4	1217	1.6×10^{-3}	1.7×10^{-4}
10000	250	2	261	1.7×10^{-2}	1.3×10^{-3}
10000	500	2	191	1.7×10^{-2}	1.3×10^{-3}
10000	350	4	290	1.5×10^{-3}	1.7×10^{-4}
10000	1150	4	289	1.5×10^{-3}	1.7×10^{-4}
10000	250	5	696	6.0×10^{-4}	4.0×10^{-5}
10000	350	5	382	6.9×10^{-4}	4.0×10^{-5}
10000	750	5	398	6.9×10^{-4}	4.0×10^{-5}
10000	1150	5	384	6.9×10^{-4}	4.0×10^{-5}
10000	1250	5	667	6.4×10^{-4}	4.0×10^{-5}
10000	1400	5	400	6.3×10^{-4}	4.0×10^{-5}
10000	250	3	380	7.0×10^{-3}	8.0×10^{-4}
10000	350	3	228	7.0×10^{-3}	8.0×10^{-4}
10000	750	3	227	7.0×10^{-3}	8.0×10^{-4}
10000	1150	3	227	7.0×10^{-3}	8.0×10^{-4}
10000	1250	3	397	6.0×10^{-3}	7.0×10^{-4}
21970	21970	1	4157	0.0×10^{-0}	0.0×10^{-0}
21970	750	4	958	1.5×10^{-3}	1.8×10^{-4}
21970	1000	4	959	1.5×10^{-3}	1.8×10^{-4}
21970	500	4	2717	1.5×10^{-3}	1.8×10^{-4}
21970	600	4	959	1.5×10^{-3}	1.8×10^{-4}
21970	1500	4	959	1.5×10^{-3}	1.8×10^{-4}
21970	2300	4	958	1.5×10^{-3}	1.8×10^{-4}
21970	2500	4	958	1.5×10^{-3}	1.8×10^{-4}

Table 2: Parallelepiped Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
21970	2900	4	1492	1.3×10^{-3}	1.8×10^{-4}
21970	2700	4	958	1.5×10^{-3}	1.8×10^{-4}
21970	500	5	3658	6.9×10^{-4}	4.4×10^{-5}
21970	750	5	1100	6.8×10^{-4}	4.4×10^{-5}
21970	2000	5	1100	6.8×10^{-4}	4.4×10^{-5}
21970	2700	5	1117	6.8×10^{-4}	4.4×10^{-5}
21970	2900	5	1636	6.2×10^{-4}	4.4×10^{-5}
68590	68590	1	38811	0.0×10^{-0}	0.0×10^{-0}
68590	1800	4	7833	1.5×10^{-3}	1.7×10^{-4}
68590	750	4	8141	1.6×10^{-3}	1.7×10^{-4}
68590	500	4	5501	1.7×10^{-3}	1.8×10^{-4}
68590	1800	5	7803	6.5×10^{-4}	4.4×10^{-5}
68590	750	5	12258	6.8×10^{-4}	4.4×10^{-5}
68590	12258	5	8808	6.8×10^{-4}	4.4×10^{-5}
92610	92610	1	75542	0.0×10^{-0}	0.0×10^{-0}
92610	750	4	7167	1.7×10^{-3}	1.8×10^{-4}
92610	750	5	10522	6.8×10^{-4}	4.4×10^{-5}

A.3 Data for Separated Cube Domain Configuration

Table 3: Separated Cube Domain Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
1000	1000	1	9	0.0×10^{-0}	0.0×10^{-0}
1000	750	4	9	3.7×10^{-3}	1.3×10^{-5}
1000	750	5	14	5.2×10^{-4}	2.8×10^{-6}
1000	100	4	13	3.7×10^{-3}	1.3×10^{-5}
4096	4096	1	139	0.0×10^{-0}	0.0×10^{-0}
4096	750	4	36	3.4×10^{-3}	1.1×10^{-5}
4096	750	3	22	2.1×10^{-2}	8.6×10^{-5}
4096	750	5	53	4.6×10^{-4}	2.0×10^{-6}
8000	8000	1	519	0.0×10^{-0}	0.0×10^{-0}
8000	750	4	80	3.3×10^{-3}	1.1×10^{-5}
8000	750	4	70	3.3×10^{-3}	1.1×10^{-5}
8000	1500	5	115	4.4×10^{-4}	3.2×10^{-6}
8000	1500	5	104	4.4×10^{-4}	3.2×10^{-6}
17576	17576	1	2483	0.0×10^{-0}	0.0×10^{-0}
17576	3000	4	153	3.2×10^{-3}	1.0×10^{-5}
17576	750	4	169	3.2×10^{-3}	1.0×10^{-5}
17576	750	5	243	4.3×10^{-4}	3.0×10^{-6}
17576	3000	5	226	4.3×10^{-4}	3.0×10^{-6}
32768	32768	1	8594	0.0×10^{-0}	0.0×10^{-0}
32768	6000	4	284	3.1×10^{-3}	1.0×10^{-5}
32768	750	4	308	3.1×10^{-3}	1.0×10^{-5}
32768	6000	5	418	4.2×10^{-4}	2.9×10^{-6}
32768	750	5	447	4.2×10^{-4}	2.9×10^{-6}
64000	64000	1	32676	0.0×10^{-0}	0.0×10^{-0}
64000	12000	4	558	3.1×10^{-3}	1.1×10^{-5}
64000	750	4	930	3.1×10^{-3}	1.1×10^{-5}

Table 3: Separated Cube Domain Configuration

N_V	N_{Vc}	N	t_{cpu} (sec)	ϵ_U	ϵ_Φ
64000	12000	5	818	4.1×10^{-4}	6.3×10^{-6}
64000	750	5	1227	4.1×10^{-4}	6.3×10^{-6}
97336	97336	1	75538	0.0×10^{-0}	0.0×10^{-0}
97336	15000	4	843	3.0×10^{-3}	9.8×10^{-6}
97336	750	4	1437	3.0×10^{-3}	9.8×10^{-6}
97336	15000	5	1269	4.0×10^{-4}	4.0×10^{-6}
97336	750	5	1845	4.0×10^{-4}	4.0×10^{-6}

(this page intentionally left blank)

Dr. Harlow G. Ahlstrom
Boeing Defense & Space Group
P.O. Box 3999, MS85-85
Seattle WA 98124-2499

Prof. Christopher Anderson
4068 Blackbird Way
Callabassi CA 91302

Analytical Methods (2)
P.O. Box 3786
Bellevue, WA 98009
Attn: F. Dvorak
B. Maskew

Prof Hassan Aref
Dept. of Applied Mech. and Eng. Sci.
University of California
La Jolla, CA 92093

Prof. Andrew S. Arena Jr..
School of Mech. and Aero. Eng.
Oklahoma State University
Stillwater, OK 74078-0545

Prof. Holt Ashley
Dept. of Aero. and Astronautics
Stanford University
Stanford, CA 94305

Prof. Bruce Bayly
Mathematics Department
University of Arizona
Tucson, AZ 85721

Dr. P. W. Bearman
Dept. of Aeronautics
Imperial College
London, SW7 2BY
ENGLAND

Prof. James Brasseur
Dept. Mech Eng.
Pennsylvania State University
University Park, PA 16802

Dr. Thomas F. Buttke
38 Marion Road West
Princeton, NJ 08540

Dr. J. Carrier
124 Me Pierre Termier
73000 Chambéry
FRANCE

Prof. A. J. Chorin
Department of Mathematics
University of California
Berkeley, CA 94720

Dr. David J. Cockrell
Department of Engineering
University of Leicester
Leicester, LE1 7RH
ENGLAND

Prof. Werner Dahm
Dept. Aerospace Engineering
University of Michigan
Ann Arbor, MI 48109-2118

Dr. M. R. Dhanak
Dept. Ocean Engineering
Florida Atlantic University
Boca Raton, FL 33431

Dr. Karl F. Doherr
Head, Applied Math. Branch
DLR, Institut für Flugmedanile
Flughafen, D33
Braunschweig
GERMANY

Dr. Feri Farassat
NASA Langley Research Center
Mail Stop 460
Hampton, VA 23681-0001

Prof. W. L. Garrard
Dept. Aerospace Eng. and Mech.
107 Akerman Hall
110 Union St. NE
Univ. of Minnesota
Minneapolis, MN 55455

Prof. Ahmed F. Ghoniem
Dept. of Mech. Eng.
Mass. Inst. of Tech.
Cambridge, MA 02139

Prof. Peyman Givi
Dept. of Mech. and Aero Eng.
State University of New York
Buffalo, NY 14260-4400

Dr. John R. Grant
Naval Undersea Center
1176 Howell Street
Building 108, Code 8233
Newport RI 02841-1708

Prof. Leslie Greengard
Courant Institute of Math. Sciences
New York University
251 Mercer Street
New York, NY 10012

Prof. G. Gregorek
Dept. of Aero. and Astronautics
Ohio State University
2070 Neil Avenue
Columbus, OH 43210

Prof. N. D. Ham
Mass. Inst. of Tech.
77 Massachusetts Ave.
Cambridge, MA 02139

Capt. Hank Helin
HQ USFA/DFAN
USAF Academy, CO 80840-5701

Prof. H. Higuchi
Dept. of Mech. & Aero. Eng.
Syracuse University
Syracuse, NY 13244

Prof. A. K. M. F Hussain
Dept. of Mech. Eng.
University of Houston
Houston, TX 77204-4792

Prof. Yassin A. Jassan
Nuclear Engineering Dept.
Texas A&M University
College Station, TX 77843

Prof. Joseph Katz
Dept. Aerospace Eng. and Eng. Mech.
San Diego State University
San Diego, CA 92182-0183

Prof. Mitsuru Kurosaka
Dept. of Aeronautics and Astronautics
Mail Stop FS-10
University of Washington
Seattle, WA 98195

Prof. Anthony. Leonard
Graduate Aeronautics Lab.
California Institute of Technology
Pasadena, CA 91125

Dr. William W. Liou
ICOMP
NASA Lewis Research Center
Cleveland, OH 44135

Prof. M. Luttges
Dept. of Aerospace Eng. Sci.
University of Colorado
Boulder, CO 80309-0429

Prof. Mark D. Maughmer
Dept. of Aerospace Eng.
Pennsylvania State University
University Park, PA 16802

Dr. Jeffery S. Marshall
Iowa Institute of Hydraulic Research
University of Iowa
300 S. Riverside Drive
Iowa City, IA 52242-1585

Dr. Eckart Meiburg
Dept. of Aerospace Eng.
University of Southern California
854 W. 36th Place
Los Angeles, CA 90089-1191

Dr. D. I. Meiron
Dept. of Applied Mathematics
California Institute of Technology
Pasadena, CA 91125

Prof. R. N. Meroney
Dept. of Civil Eng.
Colorado State University
Fort Collins, CO 80521

Prof. Philip J. Morris
Dept. of Aerospace Engineering
The Pennsylvania State University
University Park, PA 16802

NASA Johnson Space Center (2)
Attn: EG3
Houston, Tx 77058
Attn: D. B. Kanipe
R. E. Meyerson

New Mexico State University (2)
Dept. of Mech. Eng.
Las Cruces, NM 88003
Attn: Ron Pederson
G. Reynolds

Dr. Monika Nitsche
Program in Applied Mathematics
Campus Box 562
University of Colorado
Boulder, Colorado 80309-0526

Prof. Ronald L. Panton
Dept. of Mech. Eng.
University of Texas
Austin, TX 78712

Prof. Ion Paraschivoiu
Dept. of Mech. Eng.
Ecole Polytechnique
CP 6079
Succursale A
Montreal H3C 3A7
CANADA

Prof. V. Rokhlin
Department of Computer Science
Yale University
PO Box 2158
New Haven, CT 06520

Prof. Ralph Rosenbaum
School of Physics
Tel Aviv University
Ramat - Aviv, 69978
ISRAEL

Prof. P. G. Saffman
Dept. of Applied Mathematics
California Institute of Technology
Pasadena, CA 91125

Prof. William Saric
Dept of Mech. and Aerospace Eng.
Arizona State University
PO Box 876106
Tempe, AZ 85287-6106

Prof. T. Sarpkaya
Dept. Mech. Eng.
Code 69-SL
Naval Postgraduate Academy
Monterey, CA 93943

Dr. Klaus Schilling
FH Ravensburg-Weingarten
Postfach 1261
D-7987 Weingarten
GERMANY

Dr. John M. Seiner
NASA Langley Research Center
Mail Stop 165
Hampton, VA 23681-0001

Prof. J. A. Sethian
Dept. of Mathematics
University of California
Berkely, CA 94720

Dr. Karim Shariff
NASA-Ames Research Center
MS: 202A-1
Moffet Field, CA 94035-1000

Dr. David Sharpe
Department of Aeronautical Eng.
Queen Mary College
Mile End Road
London, E1 4NS
ENGLAND

Prof. Roger L. Simpson
Dept. Aerospace and Ocean Eng.
Virginia Polytechnic Institute
and State University
Blacksburg, VA 24061

Texas Tech University (2)
Dept. of Mech. Eng.
Lubbock, TX 79409
Attn: J. H. Lawrence
J. W. Oler

Dr. Boyko Tchavdarov
KaTRI
Kajima Corporation
2-19-1, Tobitakyu, Chofu-Shi
Tokyo 182, Japan

Prof. G. Trygvasson
Dept. Mech. Eng. and App. Mech.
University of Michigan
Ann Arbor, MI 48109-2125

Prof. Don H. Tucker
Department of Mathematics
University of Utah
Salt Lake City, UT 84112

University of New Mexico (2)
Dept. of Mech. Eng.
Albuquerque, NM 87106
Attn: M. S. Ingber
C. R. Truman

Dr. James Uhlman
Code 804, Bldg. 108/2
Naval Undersea Warfare Center
Newport, RI 02841-5047

U. S. Army, Natick (9)
RD&E Center
AMED/ETD
Kansas St.
Natick, MA 01760
Attn: John Caligeros
Maurice Gionfriddo
Steven Kunz
Calvin Lee
Andrew Mawn
Robert Rodier
James Sadeck
Earl Steeves
Gary Thibault

Prof. Bruce R. White
Dept. of Mech. Eng.
2052 Bainer Hall
University of California
Davis, CA 95616

Prof. C. W. Van Atta
Dept. of App. Mech. and Eng. Sci.
Mail Code 0411
University of California
La Jolla CA 92093-0411

Prof. L. Van Dommelen
Dept. of Mech. Eng.
Florida State University
PO Box 2175
Tallahassee, FL 32316-2175

Washington State University (3)
Dept. Mech. & Matl's Eng.
Pullman, WA 99164-2920
Attn: Jacob Chung
Clayton Crowe
Tim Troutt

Prof. R. E. Wilson
Dept. Mech. Eng.
Oregon State University
Corvallis, OR 97331

Dr. G. S. Winklemans
Mech. Eng. Dept.
University of Sherbrooke
Sherbrooke, Quebec
J1K-2R1
CANADA

Prof. Norman J. Zabusky
Dept. of Mech. and Aerospace Eng.
Rutgers University
PO Box 909
Piscataway, NJ 08855-0909

Dr. David Zielke
Naval Research Lab, 6-4-40
4555 Overlook Avenue, SW
Washington, DC 20375

MS0321	1400	E. H. Barsis
MS1111	1402	S. S. Dosanjh
MS1111	1421	W. J. Camp
MS1111	1421	D. W. Barnette
MS1110	1422	R. C. Allen
MS1110	1422	L. Romero
MS0819	1431	T. G. Trucano
MS0439	1434	D. R. Martinez
MS0841	1500	D. J. McCloskey
MS0836	1501	C. W. Peterson

MS0827	1502	P. J. Hommert
MS0827	1502	A. E. Hodapp
MS0827	1511	J. S. Rottler
MS0827	1511	C. E. Hickox
MS0827	1511	G. F. Homicz
MS0827	1511	S. N. Kempka
MS0827	1511	J. A. Scutt
MS0834	1512	A. C. Ratzel
MS0835	1513	R. D. Skocypec
MS0835	1513	S. R. Tieszen
MS0832	1551	W. P. Wolfe
MS0832	1551	J. K. Cole
MS0833	1552	C. E. Hailey
MS0833	1552	R. S. Baty (5)
MS0833	1552	V. L. Behr
MS0833	1552	D. D. McBride
MS0833	1552	J. M. Nelsen
MS0833	1552	J. H. Strickland (15)
MS0833	1552	D. W. Sundberg
MS0833	1552	D. E. Waye
MS0826	1553	W. L. Hermina
MS0825	1554	W. H. Rutledge
MS0825	1554	D. P. Aeschliman
MS0443	1561	H. S. Morgan
MS0437	1562	R. K. Thomas
MS0437	1562	F. J. Mello
MS0437	1562	J. W. Swegle
MS0811	1956	S. P. Goudy
MS0951	2121	R. A. LaFarge
MS0951	2121	G. A. Laguna
MS0149	4000	J. C. Cummings
MS0574	5941	D. F. Wolf
MS0704	6201	P. C. Klimas
MS0708	6214	H. M. Dodd
MS0708	6214	D. E. Berg
MS9051	8351	L. A. Rahn
MS9051	8351	W. T. Ashurst
MS9053	8366	C. M. Hartwig
MS9401	8700	R. C. Wayne
MS9043	8743	M. L. Callabresi
MS9043	8745	R. J. Kee
MS0313	9812	M. R. Vaughn
MS0411	9823	J. M. Macha
MS9018	8523-2	Central Technical Files
MS0899	7141	Technical Library (5)
MS0619	7151	Technical Publications
MS0100	7613-2	Doc. Proc. for DOE/OSTI